

---

# A Decision Support Tool for Circular Stockpile Management using Simulation

---

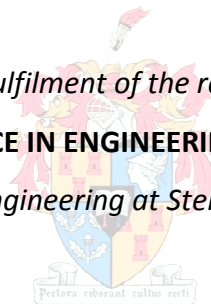
by

Zureka Loubser

*Thesis presented in partial fulfilment of the requirements for the degree of*

**MASTER OF SCIENCE IN ENGINEERING (MANAGEMENT)**

*in the faculty of Engineering at Stellenbosch University*



Supervisor: Dr Andre van der Merwe

Co-supervisor: Mr Johan de Korte (CSIR)

Department of Industrial Engineering

December 2012

## **Declaration**

By submitting this thesis/dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

**Zureka Loubser**

December 2012

## Abstract

The aim of a blending stockpile is to minimise the natural variation in properties of a material deposit. In general the reduction in variation will result in a more efficient and cost effective downstream process. Stockpile management is an important part of the production manager's duties, as it has a significant impact on the quality and consistency of the product delivered. Decisions in this regard are mostly driven by experience and specialised expertise, and a need has been identified to implement scientific methods in the decision-making process.

A representational model-driven personal decision support system was designed which aims to assist the decision making process of a production planner/manager by providing information about the expected output grade variation from a circular stockpile. The core of the decision support tool is a model that simulates the behaviour of a continuous circular blending pile. The development of the simulation model extended work done by other authors by eliminating the assumptions of constant stacker flow and vertical reclaimer slices. A user interface was developed that makes the complex calculations behind the simulation easily accessible to planning and operational personnel.

The validity of the simulation model was evaluated by using data from a case study. It was found that the coneshell model offers a reasonable representation of reality in the way that it simulates the movement of the stacker and reclaimer units around the pile, and proved able to predict peaks and troughs in the output grade to within 4% of the recorded values. Preliminary validation of the chevcon model delivered positive results, but further validation measured against recorded data would be necessary before implementation is considered.

The simulation model was also used to investigate the sensitivity of blending efficiency to various input parameters. Under optimised conditions the output variance generated by the chevcon model was half of that achieved with coneshell stacking, proving that the investment in chevcon stacking infrastructure is one that every production facility seeking improvement in output grade consistency should consider.

The research objectives set at the start of the study have been achieved, and results indicate that the decision support tool can be used to predict and the consistency of material grade as reclaimed from a circular blending pile.

## Opsomming

Die doel van 'n mengbed is om die natuurlike variasie in eienskappe van 'n mineraalafsetting te minimiseer. In die algemeen sal hierdie vermindering in variasie 'n meer doeltreffende en effektiewe proses tot gevolg hê. Mengbed bestuur is 'n belangrike deel van 'n produksie bestuurder se pligte, aangesien dit 'n merkbare invloed op die produk kwaliteit en eenvormigheid sal hê. Besluite in hierdie verband maak meestal staat op ervaring en gespesialiseerde kundigheid, maar 'n behoefte om wetenskaplike metodes in hierdie besluitnemingsproses in te sluit is egter geïdentifiseer.

'n Verteenwoordigende model-gedrewe persoonlike besluitnemingsstelsel is ontwerp, wat die produksie beplanner/bestuurder kan bystaan in hul besluitnemingsproses deur informasie te voorsien oor die verwagte uitgaande graad variasie vanaf 'n sirkelbed. Die kern van die besluitnemingsinstrument is 'n model wat die gedrag van 'n deurlopende sirkelmengbed kan simuleer. Die ontwikkeling van die simulasiemodel het uitgebrei op werk wat gedoen is deur ander skrywers deur die aanname van konstante stapelaar vloei en vertikale herwinnaar snitte te elimineer. 'n Gebruikerskoppelvlak, wat die komplekse berekeninge agter die simulaties maklik toeganklik maak vir beplannings- en bestuurspersoneel, is ook ontwikkel.

Die geldigheid van die simulasiemodelle is geëvalueer deur data vanaf 'n gevallestudie te gebruik. Die puntstapel model bied 'n redelike voorstelling van die werklikheid in die manier wat dit simuleer hoe die stapelaar en herwinnaar rondom die hoop beweeg. Die model kon ook pieke en dale in die uitgaande graad voorspel tot binne 4% van die werklike waardes. Voorlopige geldigheidstoetse op die Chevcon model het positiewe resultate gelewer, maar verdere toetse met die gebruik van werklike data sal nodig wees voor implementasie van hierdie model oorweeg kan word.

Die simulasiemodel is ook gebruik om die sensitiwiteit van vermengings effektiwiteit tot verskeie parameters te ondersoek. Onder geoptimeerde omstandighede sal die vlak van variasie soos gegenereer deur die Chevcon model die helfte soveel wees as vir die puntstapel model. Hierdie resultaat dien as bewys dat 'n belegging in Chevcon stapelaar infrastruktuur iets is wat elke produksie fasiliteit met 'n belang in verbeterde vermengings effektiwiteit behoort te oorweeg.

Die navorsingsdoelwitte soos uiteengesit aan die begin van die studie is bereik, en die resultate dui daarop dat die besluitnemingsinstrument wat ontwikkel is gebruik kan word om die eenvormigheid van materiaal, soos herwin vanaf 'n sirkelmengbed, te voorspel.

## Acknowledgements

I would like to thank everyone who contributed to my successful completion of this thesis, a few individuals in particular:

- Dr Andre van der Merwe. Thank you for your guidance and input, helping me to take this very practical subject into an academic framework.
- Mr Johan de Korte. You have been a mentor to me both inside and outside the scope of this project. Knowing that someone as knowledgeable and supportive as you are was only a phone call away gave me a great deal of comfort.
- My employer. Thank you for providing support in the form of study leave, travel arrangements, etc, and for the patience shown when my academic outputs resulted in limited work outputs.
- My colleagues and friends. I appreciated every “how are you doing on your thesis?” and “good luck with tomorrow”.
- My parents. For their unwavering confidence in me.
- Gareth. Thank you for enduring every teary phone call and sleep-deprived bad mood. Thank you for proof-reading section after section of boring stockpile-stuff. And thank you for still being around to finally celebrate this with me.

## Contents

Declaration .....	i
Abstract .....	ii
Opsomming .....	iii
List of Figures.....	viii
List of Tables .....	x
List of Equations .....	xi
Acronyms .....	xiii
1. Introduction.....	1
1.1. Background and rationale .....	1
1.2. Research problem and objectives .....	2
1.3. Research design .....	4
1.4. Research layout .....	5
2. Theoretical framework .....	6
2.1. Key technical concepts .....	6
2.1.1. Stockpile types and configurations.....	6
2.1.2. Homogenisation (Blending) .....	10
2.1.3. Material properties.....	11
2.1.4. Data recording .....	13
2.1.5. Statistical concepts .....	14
2.1.6. Mathematical concepts .....	15
2.2. The role of decision support.....	16
2.2.1. Making decisions .....	16
2.2.2. Definitions of decision support .....	17
2.2.3. Decision support tools as part of decision support.....	18
2.2.4. Classification of decision support systems .....	18
2.3. Discussion of blending literature.....	21
2.3.1. The function of a stockpile .....	21

2.3.2.	Advantages and disadvantages of different stockpiling systems.....	21
2.3.3.	Bed blending research in South Africa .....	22
2.3.4.	Sampling theory.....	22
2.3.5.	Efficiency of blending .....	22
2.3.6.	Modelling of blending piles .....	23
2.3.7.	Decision support tools in stockpile management and quality control.....	26
2.4.	Applications of bed-blending research.....	27
2.4.1.	Applications in energy coal.....	27
2.4.2.	Applications in cement .....	27
2.4.3.	Applications in iron ore .....	28
2.5.	Summary of blending literature discussion .....	29
3.	Research design.....	31
3.1.	Research approach .....	31
3.2.	Development of a simulation model .....	34
3.3.	Model validation.....	35
3.4.	Sensitivity analysis .....	35
4.	Simulation model development .....	38
4.1.	Geometric modelling .....	38
4.1.1.	Coneshell stacking model .....	38
4.1.2.	Chevcon stacking model .....	41
4.2.	Stacking with variable feed rate .....	41
4.2.1.	Coneshell method: Stacking the first cone.....	44
4.2.2.	Coneshell stacking: Stacking consecutive shells.....	49
4.2.3.	Chevcon stacking: Stacking layers .....	53
4.2.4.	Drawing the stockpile .....	59
4.3.	Reclaiming at an angle.....	60
4.4.	Forecasting reclaimed grades.....	63
4.5.	User interface .....	64
4.5.1.	Coneshell stacking .....	64

4.5.2.	Chevcon stacking .....	67
5.	Model validation.....	68
5.1.	Coneshell model .....	69
5.1.1.	Accuracy of geometric modelling.....	69
5.1.2.	Accuracy of grade forecast .....	70
5.2.	Chevcon model .....	74
5.2.1.	Mass balance .....	74
5.2.2.	Average grade.....	75
5.2.3.	Analysis of grade predictions.....	75
6.	Sensitivity analysis .....	77
6.1.	Coneshell stacking .....	77
6.1.1.	Stacker step increment.....	77
6.1.2.	Stacked height .....	78
6.2.	Chevcon stacking .....	78
6.2.1.	Rotation angle .....	78
6.2.2.	Stacker speed.....	79
6.2.3.	Number of layers / step increment .....	80
6.3.	Coneshell vs. Chevcon stacking .....	81
7.	Model application.....	83
7.1.	Operations research .....	83
7.2.	Operations management.....	83
7.3.	Financial management .....	85
8.	Conclusions.....	86
	References .....	89
	Appendix A: Simulation results .....	93



## List of Figures

Figure 1: Full-face reclaimer cutting into a stockpile .....	3
Figure 2: Variation of stacker feed .....	4
Figure 3: Circular stockpile showing stacker and reclaimer in operation .....	6
Figure 4: A stockpile being stacked using the coneshell method.....	7
Figure 5: Layers formed by chevron stacking.....	7
Figure 6: Chevcon stacking - a top view of a circular pile .....	8
Figure 7: Comparison of stacking methods - Adapted from (Bond, Coursaux & Worthington, 2000) .....	9
Figure 8: Full-face reclaimer device.....	10
Figure 9: Measuring the repose angle of material .....	12
Figure 10: Representation of particle size segregation in stockpiling operations (Mikkelsen, 1983).....	12
Figure 11: Trajectory segregation.....	13
Figure 12: Polar coordinates – Adapted from (Stewart, 2003) .....	15
Figure 13: Process steps in the manufacture of Portland cement – Adapted from (Carpio, Coelho, Silva & Jorge, 2005) .....	28
Figure 14: Target composition of iron ore for steel production.....	28
Figure 15: Components of a model-driven DSS.....	32
Figure 16: Block model on Cartesian axes.....	34
Figure 17: Two-dimensional representation of how blocks are filled.....	35
Figure 18: Counter numbers for Radial, Angular, and Height arrays .....	40
Figure 19: Flowchart of ReadStackTonnages macro .....	43
Figure 20: Model output - stacking the first cone .....	45
Figure 21: Height of a cone given distance from midpoint .....	46
Figure 22: Assigning grade values to the first cone.....	46
Figure 23: Model output - stacking successive cones .....	47
Figure 24: Blocks filled with grades as successive conical shells are stacked .....	48
Figure 25: Output of model after stacking consecutive coneshells .....	49
Figure 26: Method for calculating the intersecting volume of two cones .....	49
Figure 27: Two cones intersecting.....	51
Figure 28: GOALSEEK method executed to find shell radius.....	52
Figure 29: Output of model after stacking chevcon layers .....	54
Figure 30: Flowchart of StackLayers macro.....	55
Figure 31: Formation of chevcon volume increments .....	57
Figure 32: Calculation of height of a chevcon layer .....	58
Figure 33: Output of reclaimer model.....	60

Figure 34: Calculation of slice height.....	62
Figure 35: Coneshell stacking full graphic user interface .....	64
Figure 36: Coneshell stacking GUI Part 1.....	65
Figure 37: Coneshell stacking GUI Part 2.....	66
Figure 38: Coneshell stacking GUI Part 3.....	66
Figure 39: Chevcon stacking GUI Part 2.....	67
Figure 40: Accuracy of stacker geometric modelling .....	69
Figure 41: Accuracy of reclaimer geometric modelling.....	70
Figure 42: Coneshell model - Accuracy of grade forecast .....	71
Figure 43: Coneshell model - Accuracy of adjusted grade forecast .....	72
Figure 44: Coneshell stacking - Accuracy of additional adjusted grade forecast .....	73
Figure 45: Chevcon model - Accuracy of grade forecast.....	76
Figure 46: VRR plotted as a function of stacker increment.....	77
Figure 47: VRR plotted as a function of stacked height .....	78
Figure 48: VRR plotted as a function of blending tail length.....	79
Figure 49: VRR plotted as a function of stacker speed.....	80
Figure 50: VRR plotted as a function of layer increment .....	81
Figure 51: Simple production planning decision making process .....	84

## List of Tables

Table 1: Representation of bed-blending model.....	24
Table 2: Identification of suitable DSS types .....	33
Table 3: Simulations performed - coneshell model.....	36
Table 4: Simulations performed - chevcon model .....	37
Table 5: Constant properties used as input to the geometric model .....	39
Table 6: Quadrant adjustment in CreatePoints.....	41
Table 7: Coneshell validation - Statistical parameters for modelled and recorded data.....	71
Table 8: Coneshell stacking - Correlation between modelled and recorded data .....	73
Table 9: Mass balance over validation simulations.....	74
Table 10: Average grade stacked and reclaimed for validation simulations.....	75
Table 11: Chevcon validation - Statistical parameters for modelled and recorded data.....	76
Table 12: Comparison of coneshell and chevcon stacking .....	81
Table 13: Simulation results for coneshell model .....	94
Table 14: Simulation results for chevcon model .....	95

## List of Equations

Equation 1: Calculation of the variance reduction ratio (Kumral, 2006) .....	10
Equation 2: Calculating the angle of repose when height and base width are known .....	11
Equation 3: The limiting distance a particle can travel horizontally (Rhodes, 1998) .....	13
Equation 4: Calculation of standard deviation (Weisstein, 2012b) .....	14
Equation 5: Finding $r$ in polar coordinates from Cartesian coordinates (Stewart, 2003) .....	15
Equation 6: Finding $\theta$ in polar coordinates from Cartesian coordinates (Stewart, 2003) .....	15
Equation 7: Finding $x$ in Cartesian coordinates from polar coordinates (Stewart, 2003) .....	15
Equation 8: Finding $y$ in Cartesian coordinates from polar coordinates (Stewart, 2003) .....	15
Equation 9: Distance between two points in the Cartesian plane (Weisstein, 2012a) .....	16
Equation 10: Distance formula in polar coordinates .....	16
Equation 11: The volume of a cone .....	16
Equation 12: Calculation of sampling error variance (Robinson, 2004) .....	22
Equation 13: Minimum number of stacked layers for proper blending (SACPS, 2011) .....	23
Equation 14: Estimation of the homogenising effect of a blending pile (De Wet, 1994) .....	23
Equation 15: Trigonometric relationship between the cone radius, angle of repose, and stockpile height ..	39
Equation 16: Calculation of a circle's chord length when radius and central angle are known .....	44
Equation 17: Cone volume expressed in terms of radius and repose angle .....	45
Equation 18: Radius of a cone when volume is known .....	45
Equation 19: Height of a cone given distance from midpoint .....	46
Equation 20: Volume of a shell over a smaller cone .....	47
Equation 21: Radius of conical shell .....	47
Equation 22: Determination of intersection cone radius .....	50
Equation 23: Relationship between $a$ and $b$ .....	50
Equation 24: Calculation of $a$ .....	50
Equation 25: Calculations in ConeShellVolume .....	50
Equation 26: Calculation of coneshell volume .....	51
Equation 27: Target value for coneShellVolume .....	52
Equation 28: Length of a stacker pass in chevcon mode .....	53
Equation 29: Calculation of time per stacker pass .....	53
Equation 30: Movement increment angle of the stacker boom during chevcon mode (Holderbank, n.d.) ...	53
Equation 31: Calculation of increment height .....	54
Equation 32: Calculation of target volume .....	56
Equation 33: Calculation of increment volume .....	56
Equation 34: Volume of a chevcon layer .....	57

Equation 35: Calculation of volume parameters.....	57
Equation 36: Calculation of base width for a chevcon layer .....	58
Equation 37: Calculation of height of a chevcon layer .....	58
Equation 38: Calculation of reclaimer length angle .....	61
Equation 39: Calculation of slice height .....	62
Equation 40: Calculation of a slice volume.....	63
Equation 41: Calculation of reclaimer length.....	79

## Acronyms

BI	–	Business Intelligence
DS	–	Decision Support
DSS	–	Decision Support System(s)
EIS	–	Executive Information System(s)
GIGO	–	Garbage In Garbage Out
GSS	–	Group Support System(s)
GUI	–	Graphic User Interface
IDSS	–	Intelligent Decision Support System(s)
KM	–	Knowledge Management
NSS	–	Negotiation Support System(s)
PC	–	Personal Computer
PDSS	–	Personal Decision Support System(s)
VBA	–	Visual Basic for Applications
VRR	–	Variance Reduction Ratio

## 1. Introduction

### 1.1. Background and rationale

Even in the most uniform deposit of materials in nature, material properties will still vary. The degree of this variation can be measured, usually resembling a normal distribution. The aim of blending and homogenisation operations is to minimize the standard deviation of this normal distribution (De Wet, 1994). In general the reduction in variation will result in a more efficient and cost effective downstream process. Cement plants need a precise blend of raw materials to ensure desired kiln performance, boilers used in power generation are optimised for fuels of a constant specification, and even though the separation density in a coal washing plant can be adjusted, a constant product quality cannot be produced from highly variable feed (Denny & Harper, 1962). The process efficiency, product quality, and environmental compliance depend on consistency of characteristics in the material fed (Kumral, 2005).

Production facilities strive for consistency in their products, be it cement, electricity, or coal. Consistency can be defined as an “agreement, harmony, or compatibility, especially correspondence or uniformity among the parts of a complex thing” (Dictionary.com, 2012). Homogenisation of the feed to a production plant can ensure consistency of the product in all respects (Denny & Harper, 1962).

Many industries involved in chemical or minerals processing require a level of raw material homogenisation as part of their process. Product homogenisation can also be needed if the process yields a product that is not sufficiently uniform (Van der Mooren, 1962). One of the most widely used methods of homogenisation is stockpiling. Copper plants in Central and South America were some of the first to start using bed-blending as a method of feed homogenisation (Denny & Harper, 1962).

Blending is not the only function of a stockpiling system. Stockpiles act as buffers between two processes, so that either can continue without being constrained by the other. More often than not the buffering function provides the primary economic consideration in stockpile design, so when homogenisation options are evaluated only minor adjustments to the existing system can be considered (Robinson, 2004). If a consistently blended feedstock is provided, the following advantages can be expected (De Wet, 1983):

- Stable process operation, resulting in lower operating costs and higher product quality
- The process plant can be optimally sized for a given throughput rate, as the grade of feedstock can be more accurately predicted
- Product yield is higher, therefore raw material consumption (and resultantly costs) is lower
- Product quality is controlled within smaller tolerances. This has cumulative advantages, as the product from one process is often used as input to another

Stockpile management is an important part of the production manager's duties, as it has a significant impact on product quality and consistency. Decisions in this regard are mostly driven by experience and specialised expertise, and a need has been identified to implement scientific methods in the decision-making process.

## 1.2. Research problem and objectives

The management of a blending pile involves many decisions, some made on a day-to-day or even hour-to-hour basis.

- "Taking into account the material that is currently on the pile, will adjustments to the downstream process be necessary?"
- "If we were to stack an incoming source of a different type, what influence would it have on the pile's internal variation?"
- "What can we expect if we blend the output from this stockpile with another source?"
- "Could increasing the stacking speed yield significant improvement in the consistency of the pile's output?"

These decisions, and many more like them, can be made more effectively if reliable information about the stockpile variation is available. Ideally the production manager/planner would want to know as accurately as possible what kind of material is on the pile, and where it is distributed. Unfortunately this type of information is not routinely available, which means that decisions are made on the basis of estimation and "gut feel".

In order to create a three-dimensional profile of a circular stockpile, which could possibly be used to provide the information necessary for decision-making, one would need to simulate how material is stacked and reclaimed from the pile. Although much work has been done on the simulation of blending (Gerstel, 1989; Gerstel, 1979; Hurwitz & Ackermann, 1999; Kumral, 2006; Pavloudakis & Agioutantis, 2003; Robinson, 2004), it has been focussed mostly on linear stockpiles. Modelling is often of a statistical nature, which requires some prior knowledge of the input grade variation. Statistical models are also based on many assumptions, some of which are not applicable to most material handling applications.

No simulation model could be found which accurately estimates the internal variation of material and output characteristics of a circular stockpile when little is known about the input characteristics. The available literature body also lacks a comparison between the blending efficiencies that can be achieved between different stacking methods on a circular pile, although comparisons have been reported for linear stacking methods.



From a perusal of the literature available it is clear that the simulations documented so far used a degree of assumption and approximation in the model development. This will of course always be the case in modelling and simulation. Although these approximations are for the most part an accurate enough indication, they could be adjusted to better reflect real-life situations. This research project will aim to eliminate two important approximations.

Firstly, almost all of the published simulations discussed used a vertical “slice” to model the action of a full-face reclaimer. In reality the reclaimer is cutting into the stockpile at an angle, usually equal to the angle of repose of the material (see Figure 1). Therefore the proportions of each stacked layer represented in a reclaimed section will be significantly different than predicted for a vertical cut.



Figure 1: Full-face reclaimer cutting into a stockpile

Secondly, all the simulations mentioned assume a constant material flow from the stacker, which is rarely the case in practice. Figure 2 shows tonnes sent to the stacker boom conveyor at a coal production facility, plotted every hour for a month. It is clear that the rate of material flow from the stacker will be highly variable. Modelling of the stacked layers can be made more accurate by accommodating variation in the material feed rate.

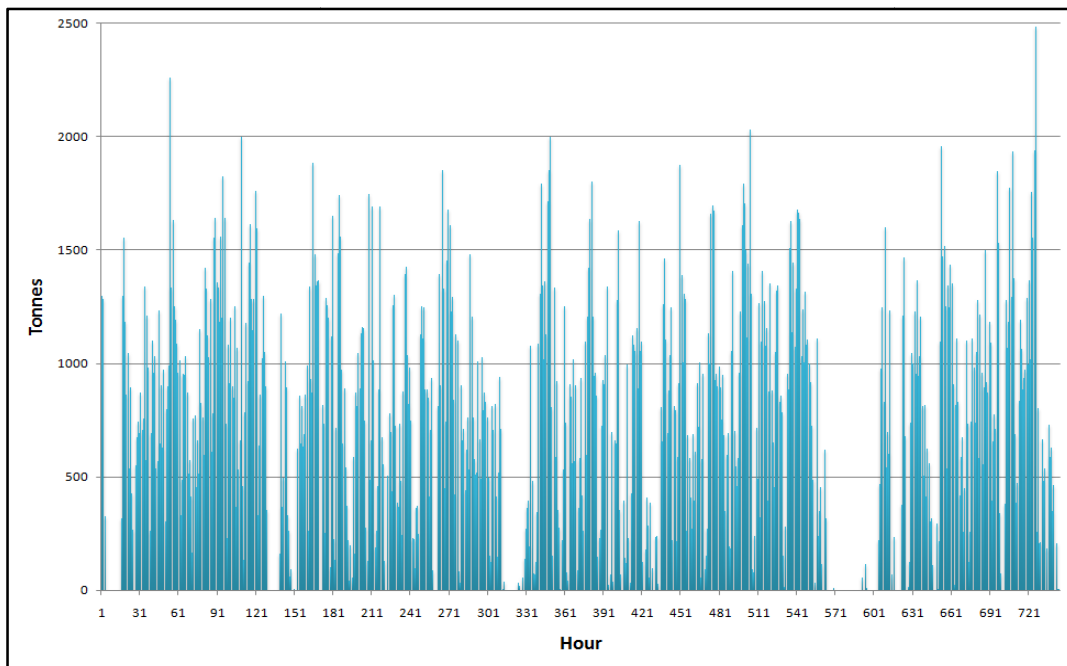


Figure 2: Variation of stacker feed

The purpose of this project is to bridge the gaps identified in solids handling research by developing a simulation model that can be used to profile the material variation in a continuous circular blending pile. The simulation model will then be used to provide information in the form of a decision support tool for managing circular stockpiles. Not only will this tool enable the user to predict the properties of material that will be reclaimed and plan accordingly, but also to estimate the blending efficiency that can be achieved for a given set of operating parameters. The tool must be easy to use with minimal training, enabling stockpile management and planning personnel to make informed decisions by using data that is already available.

### 1.3. Research design

The main research question that this project seeks to answer is: “Can variation of material properties as reclaimed from a circular stockpile be predicted?”. Secondary objectives that will assist in answering the research question are as follows:

- ✓ Create a geometric model that can accurately represent a circular stockpile stacked in either the chevcon or the coneshell method.
- ✓ Incorporate variable feed rate in the way the stacker model forms layers.
- ✓ Simulate how material will be reclaimed from the stockpile by slicing into it at an angle.
- ✓ Develop a user interface for the simulation model. The interface must be easily navigated by users, displaying all information necessary for day-to-day decision making.
- ✓ Test the model with known input and output data to evaluate how accurately it represents reality.
- ✓ Evaluate the effects of varying operational parameters by using blending efficiency as a measure.
- ✓ Quantify the difference in blending efficiency for chevcon and coneshell stacking methods.

## 1.4. Research layout

Chapter 1 has provided the background to the research problem, and aimed to emphasize the importance of bed blending research. The research problem and the objectives that this research project set out to achieve were identified.

In Chapter 2 key technical concepts that are important to the reader's understanding of the research project are defined. Furthermore the theoretical background associated with the field of decision support is mapped by using frameworks from published works. This is followed by a discussion of relevant blending literature and the application of bed-blending research to different commodities. The chapter concludes with a summary of some important points that came to light during the literature review.

Chapter 3 details the research design and methodology followed. Decision support theory is put into practice by evaluating different options for development of a decision support tool at the hand of design criteria. The chapter also serves as introduction to Chapters 4 to 6, by briefly explaining how the simulation model was designed and evaluated.

Chapter 4 is a detailed, step-by-step discussion of how the simulation model was developed. It provides explanations of the VBA code shown in the appendix, by defining all variables and macros used. All assumptions and approximations used in the model development are detailed here. Lastly the user interface is presented and discussed.

Chapter 5 evaluates how accurately the simulation model represents reality. It makes use of data from a case study to plot recorded versus modelled behaviour and draw conclusions as to the simulation model's accuracy.

In Chapter 6 the sensitivity of model output to certain input parameters is evaluated. It serves to demonstrate the effect that a difference in the way a stockpile is operated can have on the consistency of the output material.

Chapter 7 discusses possible applications of the simulation model, and associated decision support tool, within a broader management science context.

The conclusions drawn from the final chapters are summarised in Chapter 8.

## 2. Theoretical framework

### 2.1. Key technical concepts

#### 2.1.1. Stockpile types and configurations

All stockpiles involve stacking material (feeding *to* the pile) and reclaiming it (feeding *from* the pile) at a later stage. There are many different stockpile configurations in operation, with varying geometries, stacking methods, and reclaiming methods. Only the configurations most applicable to this study will be discussed below.

##### *Linear stockpiles*

A linear homogenisation system always consists of at least two piles. The stacker mostly travels from one end of the pile to the other, while the reclaimer works on only one end of the pile. In order to avoid interference between these two machines, one pile is always being stacked while the other is being reclaimed. A major disadvantage of linear stockpiles is the high variation in material properties at the ends of the piles, known as the “end-cone effect”. Robinson (2004) proved with the use of modelling that the first and last sections reclaimed from a linear stockpile have an over-representation of the last-stacked material, as it is the material on the outside of the stockpile at the time.

##### *Circular stockpiles*



Figure 3: Circular stockpile showing stacker and reclaimer in operation

The stacker and reclaimer rotate about a central axis in the same direction, with one end being stacked while the other is reclaimed. The operation is therefore continuous, which is why circular stockpiles pose the advantage of not exhibiting the end-cone effect. Circular piles have been used worldwide since the late 1970s, notably in the coal, steel, and cement industries (Gerstel, 1989). Numerous installations are currently operating in South Africa (De Wet, 1983, 1994).



### *Coneshell stacking*

Coneshell stacking is the most common method for storage systems that do not have a homogenisation purpose (Wintz, 2011). The stacker starts discharging material at a single point, forming a cone. It then moves incrementally to form successive cones in shells over the first cone. Coneshell stacking can be used on linear or circular stockpiles.



Figure 4: A stockpile being stacked using the coneshell method

### *Chevron stacking*

The stacker travels at almost constant speed along the full length of the pile, back and forth, continuously discharging material. The first pass will form a small triangular prism, and every successive pass will stack a layer with the cross-section of an inverted “V” on top of it, as shown in Figure 5. If the feed rate is constant the thickness of layers diminishes as the pile is built, until a full cross-section is formed.

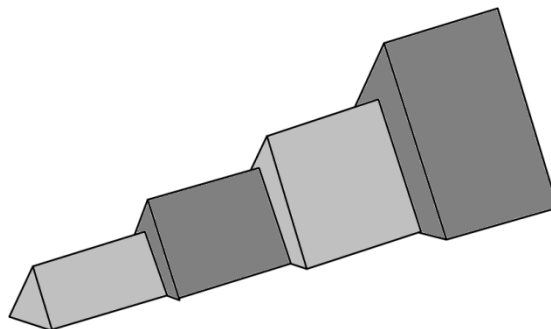


Figure 5: Layers formed by chevron stacking

### *Chevcon stacking*

Chevcon stacking is similar to chevron stacking, but designed specifically for use on circular piles. The stacker moves back and forth within a preset angle of rotation in a slewing motion. At the same time the stacker moves up and down in a luffing motion from the top of the existing stockpile to ground level, forming a blending tail. The angle of incline chosen for this motion needs to be the same or smaller than the angle of repose of the material (see section 2.1.3), and the length of the blending tail is determined by the homogenisation requirements (Hurwitz & Ackermann, 1999).

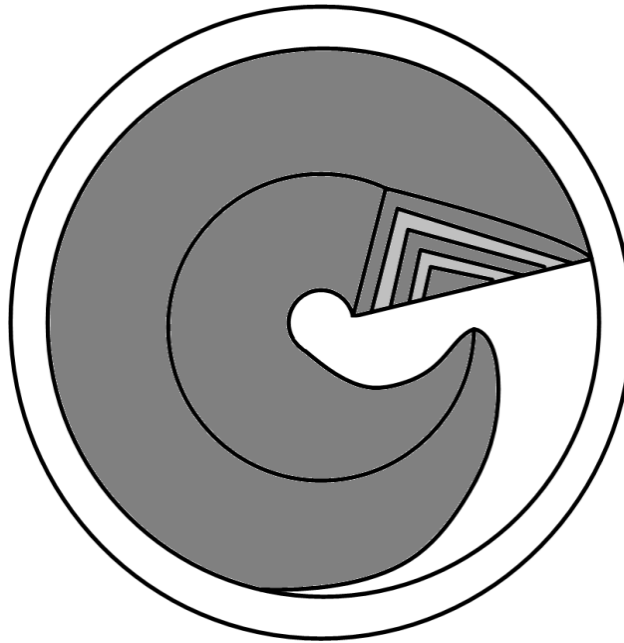


Figure 6: Chevcon stacking - a top view of a circular pile

### *Windrow stacking*

Stacking is performed by traversing the length of the stockpile on different axial lines, therefore forming several parallel triangular prisms on every level. After the first level of windrows is stacked, the second level is stacked in the spaces between the peaks of the first level. This process is repeated until the desired stockpile height is reached. Windrow stacking is only suitable for use with linear stockpiles. It is especially useful in applications with highly variable particle sizes, as it better avoids grain size segregation during stacking (Bond, Coursaux & Worthington, 2000). This is an expensive stacking method though, as it requires the use of a stacker that can perform both slewing and luffing motions (Mikkelsen, 1983).

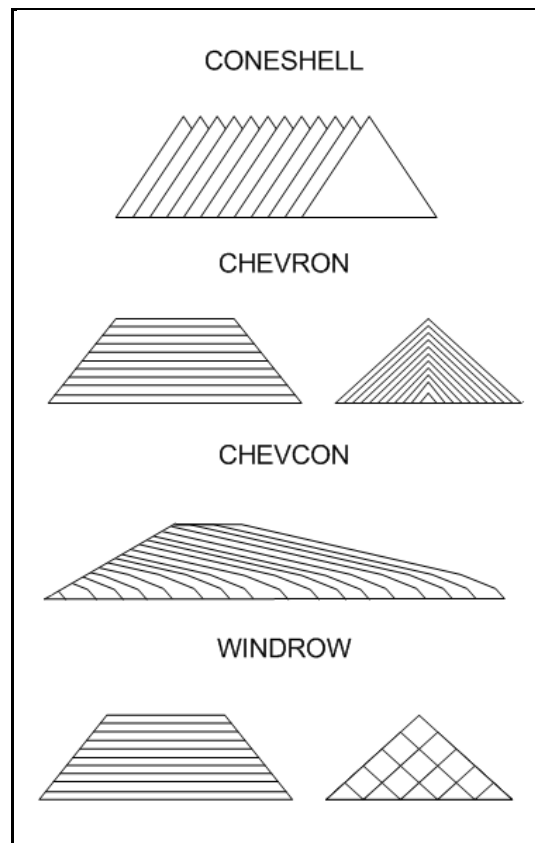


Figure 7: Comparison of stacking methods - Adapted from (Bond, Coursaux & Worthington, 2000)

### *Full-face reclaiming*

The reclaiming device consists of a harrow (also called a rake) large enough to cover the cross-section of the pile. The harrow is positioned at the angle of repose of the material, but can be varied slightly. During operation it is moved from side to side, disturbing the face of the pile. Particles break away from the face and cascade to the bottom of the harrow, where it is collected and discharged onto a conveyor. Figure 8 shows a full face reclaimer device operating on a circular stockpile, with the harrow visible on a cross-section of the pile. Buckets mounted on a chain (visible in the lower half of Figure 8) scrape reclaimed material towards the central column, from where it passes through a chute onto the reclaimer conveyor. The reclaimer is mounted at its inner end on a slewing ring, and on rail mounted drive bogies on its outer end. This is to ensure that the reclaimer rake is always confronted by a full stockpile cross-section (Wolpers, 1995).

Since the cross-section of the pile will consist of increments from different stacked layers, the discharged material is a blend of the stacked material. Reclaiming using a full-face reclaimer results in the highest homogenisation efficiency (SACPS, 2011).



Figure 8: Full-face reclaimer device

### *Bench reclaiming*

The stockpile is reclaimed in horizontal layers, by using a reclaimer that moves in a sickle-like slewing motion. Bucket wheel reclaimers are generally used for this type of reclaiming. As the name suggests, a rotating wheel with buckets attached is fitted at the end of the reclaimer boom, and is used to gather material that is later discharged onto a conveyor. Bench reclaiming has generally been shown to deliver poor blending performance, and the problem is compounded in windrow stacking, as the benches are further from being an equal mixture of all input blocks. If a boom reclaimer takes a large amount of material from a bench before moving on to the next one the variation between benches will dominate the variation in the output (Robinson, 2004).

#### **2.1.2. Homogenisation (Blending)<sup>1</sup>**

A homogeneously blended pile is one that has the same composition throughout, of which the properties of any smaller sample of material will apply to the pile as a whole. Homogenisation implies that the fluctuations of a property in the input flow are smoothed in the output, which results in a reduced standard deviation (Gerstel, 1989). The degree of homogeneity can be expressed as the standard deviation of a material property relative to its mean value (Denny & Harper, 1962). One way of measuring blending or homogenisation efficiency is the variance reduction ratio (VRR), where:

$$VRR = \frac{\sigma_{out}^2}{\sigma_{in}^2}$$

Equation 1: Calculation of the variance reduction ratio (Kumral, 2006)

<sup>1</sup> “Homogenisation” and “Blending” are used interchangeably throughout stockpile literature, and also in this document



In Equation 1,  $\sigma_{out}^2$  and  $\sigma_{in}^2$  are the output and input variances respectively. Input and output variances should be calculated on the same base, i.e. identical weights or volumes.

The efficiency of the blending system is dependent on three factors (Kumral, 2006):

- Stockpiling method. This is the way in which material is placed onto the stockpile by the stacker. The stacker is mostly responsible for homogenisation, not the reclaimer (SACPS, 2011).
- Stockpiling parameters. These include the length, width, number of layers stacked, equipment properties of the stacker and reclaimer, and raw material characteristics, among others.
- Input variability. The frequency and amplitude of the variation of material properties in the input stream.

### 2.1.3. Material properties

#### *Material grade*

Stockpile modelling as discussed in this framework can be used to predict any additive material property, but mostly the consistent “grade” of material will be the main driver for plant optimisation. Grade can refer to any one of a range of material properties, depending on the material being processed and the downstream application.

#### *Angle of repose*

The angle of repose can be defined as the slope formed between the horizontal and a heap of material dropped from a known height (McGlinchey, 2008). The angle at which the pile forms is determined by the internal shear force between particles.

Two standards exist for measuring the angle of repose of a material, the British Standard 4140-9 of 1986 and ISO 4327 of 1977. The method involves discharging material from a funnel with a cut-off stem, mounted on a tripod at known height, onto a plate. The plate is inscribed with circles, each marked with an angle of repose corresponding to the ratio between height and circle radius. Alternatively the angle of repose can also be determined visually. Figure 9 demonstrates the method, corresponding to an angle of repose of  $36.5^\circ$ .

Equation 2 can also be used to calculate the angle of repose after the pile height and base width are measured.

$$\text{angle of repose} = \arctan\left(\frac{2 \times \text{height}}{\text{base}}\right)$$

Equation 2: Calculating the angle of repose when height and base width are known



Figure 9: Measuring the repose angle of material

The angle of repose is not a set property for any given material, as it is dependent on a number of factors. The distribution of particle size will play a role, as a pile of large rocks will not behave in the same way as a heap of powder. The material moisture content is also important, as this will determine to what extent particles stick to each other. Therefore higher moisture content will result in larger repose angles, as material will be less likely to roll down the side of the pile.

### *Bulk density*

Bulk density is measured as the tonnes of material packed in a cubic meter. Its value is determined mainly by the particle density and the way that the material stacks into a space, i.e. how much space is left between particles. Bulk density is therefore highly dependent on particle size. As a result of the voids between particles the bulk density is always less than the particle density.

### *Segregation*

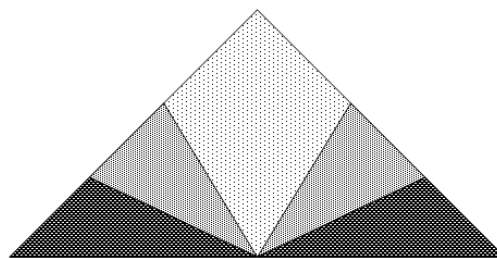


Figure 10: Representation of particle size segregation in stockpiling operations (Mikkelsen, 1983)

Segregation, in this context, refers to the way that different particle sizes end up in different parts of the stockpile. This is especially a problem when material grade varies with grain size. Two types of segregation are common in stockpiling operations. The first is caused by larger materials rolling downhill to the side of the stockpile, but smaller particles staying at the top. A grain size profile as shown in Figure 10 can be expected.

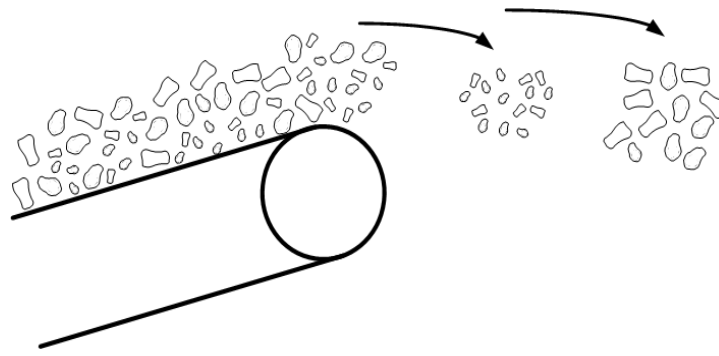


Figure 11: Trajectory segregation

The second mechanism of segregation is called trajectory segregation (see Figure 11). According to (Rhodes, 1998) the limiting distance that any particle can travel horizontally is described as follows:

$$distance = \frac{U\rho_p x^2}{18\mu}$$

Equation 3: The limiting distance a particle can travel horizontally (Rhodes, 1998)

In Equation 3  $x$  is the particle diameter,  $\rho_p$  is the particle density,  $U$  is the initial velocity, and  $\mu$  is the fluid (air, in this case) viscosity. All particles are given the same initial velocity after being discharged from the conveyor belt, and they travel through the same fluid. Segregation will therefore occur according to differences in density and particle size. A particle that is twice as large will travel four times as far, which is why larger particles tend to be found on the side of the stockpile furthest from the stacker discharge.

The effect of segregation can be largely cancelled by making use of reclaimers equipped with raking devices which scrape the full cross-section of the pile (Mikkelsen, 1983).

#### 2.1.4. Data recording

##### *Tonnages*

The material flow to and from a stockpile is mostly measured by scales placed on the relevant conveyor belts. The scales feed real-time information into the control system, and records tonnages per time increment in a database.

##### *Material grade*

Traditionally the data used to describe material properties stacked to or reclaimed from a blending pile would be obtained by incremental sampling of the input/output streams. Sampling introduces a large element of human error, and a time-lag in processing of the results. With modern technology, however, on-line analysis of many of these properties is now possible, meaning that accurate real-time monitoring of blending performance can be achieved.

### *Stacker and reclaimer positions*

At the production facility used for the case study in chapter 5, stacker and reclaimer positions are noted manually. The stockpile is divided into radial sectors (as on a wagon wheel), and each sector numbered. The numbers are painted on the side of the pile, all around the circumference. A field operator notes the position of the stacker and reclaimer every two hours by looking in which sector they are, and reporting back to the control room operator. The control room operator in turn notes these positions on a quality control sheet. This system is unfortunately largely reliant on the judgement and competency of the operators involved. Positions might not be noted for every interval, or they might be noted incorrectly. Also, because the painted sector numbers are quite far apart, it is sometimes difficult for the field operator to decide which sector number to use when the stacker or reclaimer is between two numbers. The problem is compounded by the fact that different shifts are involved, meaning that rules of thumb used to overcome these difficulties aren't consistently applied.

Modelling of a blending pile can be made much more accurate if automation of data entry is used. Technology is available to track the stacker and reclaimer positions and feed automatically into the control system and/or quality control sheet. Discussion of the way this technology works is outside the scope of the research done, but it is something to keep in mind when looking at ways to increase the accuracy of predictions and forecasts made.

#### **2.1.5. Statistical concepts**

##### *Standard deviation*

Standard deviation is a measure of the intensity of fluctuations around the mean of a property (Gerstel, 1989). It can be calculated as follows:

$$S = \sqrt{\frac{\sum_{i=1}^n a_i^2}{n} - \mu^2}$$

**Equation 4: Calculation of standard deviation (Weisstein, 2012b)**

In Equation 4  $S$  is the standard deviation,  $a_i$  is the value of data point  $a$ ,  $n$  is the number of data points, and  $\mu$  is the mean value of the data set. The empirical rule of statistics states that, if data follows a normal distribution, 99.7% of values will fall within three standard deviations of the mean, 95% of data will fall within two standard deviations, and 68% within one.

### 2.1.6. Mathematical concepts

#### *Cartesian vs. Polar coordinates – from (Stewart, 2003)*

A coordinate system is a way of representing a point in space, by assigning a set of numbers called coordinates. Cartesian coordinates, used conventionally, are distances from perpendicular axes. For a point P its two-dimensional Cartesian coordinates would be given as  $P(x,y)$ . Newton introduced another coordinate system which is more convenient to use for some purposes, namely the polar coordinate system. The polar coordinate system will be demonstrated by use of Figure 12.

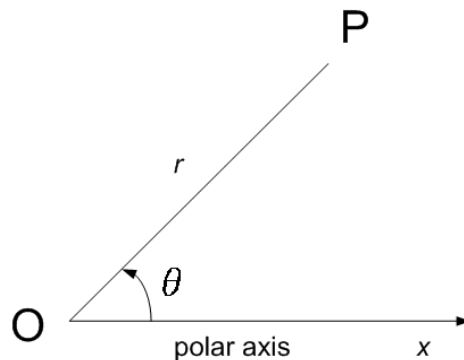


Figure 12: Polar coordinates – Adapted from (Stewart, 2003)

The point labeled “O” is called the pole, or the origin. A line is drawn from O, horizontally to the right, which corresponds to the positive x-axis in Cartesian coordinates. This is called the polar axis. Then  $r$  is the distance from O to P, and  $\theta$  is the angle between OP and the polar axis, measured in radians. The point P can be expressed in polar coordinates as  $P(r, \theta)$ .

Conversion from Cartesian coordinates to polar coordinates can be done by using Equation 5 and Equation 6 below.

$$r = \sqrt{x^2 + y^2}$$

Equation 5: Finding  $r$  in polar coordinates from Cartesian coordinates (Stewart, 2003)

$$\theta = \arctan\left(\frac{y}{x}\right)$$

Equation 6: Finding  $\theta$  in polar coordinates from Cartesian coordinates (Stewart, 2003)

Similarly one can convert from polar coordinates back to Cartesian coordinates by using Equation 7 and Equation 8.

$$x = r \cos\theta$$

Equation 7: Finding  $x$  in Cartesian coordinates from polar coordinates (Stewart, 2003)

$$y = r \sin\theta$$

Equation 8: Finding  $y$  in Cartesian coordinates from polar coordinates (Stewart, 2003)

### *Distance between two points*

The distance between two points in the Cartesian plane can be found by using Equation 9.

$$d = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$$

Equation 9: Distance between two points in the Cartesian plane (Weisstein, 2012a)

Substituting Equation 7 and Equation 8 the formula for the distance between two points in polar coordinates is as follows:

$$d = \sqrt{r_1^2 + r_2^2 - 2r_1r_2\cos(\theta_1 - \theta_2)}$$

Equation 10: Distance formula in polar coordinates

### *Volume of a cone*

When modelling how material is stacked in the coneshell method, the volume of a cone needs to be calculated frequently. The formula used is shown in Equation 11.

$$V = \frac{1}{3}\pi r^2 h$$

Equation 11: The volume of a cone

## **2.2. The role of decision support**

### **2.2.1. Making decisions**

A decision is defined as the choice of one among a range of alternatives (Bohanec, 2003). Decision making concerns the process whereby decisions are made, selecting an alternative so as to best satisfy the aims or goals of the decision maker (Efsthathiou and Rajkovic, 1979).

According to Bohanec (2003), the decision making process includes:

- i. Assessing the problem
- ii. Collecting and verifying information
- iii. Identifying alternatives
- iv. Anticipating consequences of decisions
- v. Making the choice using sound and logical judgement based on available information
- vi. Informing others of decision and rationale
- vii. Evaluating decisions

This research is focussed on step ii, where the information needed to choose between alternatives is gathered. Decision-makers receive and analyze information using different media, including traditional print, interpersonal information exchanges, and computer-based tools (Power, 2001). When the right information is not available people will utilise rules-of-thumb and shortcuts to formulate judgements and choose among alternatives (INSEAD, 2001).

### 2.2.2. Definitions of decision support

Decision support is a broad term that refers to all aspects related to supporting people in making decisions (Bohanec, 2003). Decision support is defined in a variety of different ways, depending on the context. Definitions found in literature that were deemed relevant to this study are listed below.

- “Identifying all the data required to make a decision, gathering it together organised as meaningful information” (Morrison & Moore, 1999)
- “Structured, sometimes mathematically based, approaches to decision making” (Gilfillan, 1997)
- “DS means helping you to make good decisions by understanding the effects of all the alternatives” (SRI online, 2001)
- “Specialised type of data analysis developed to enhance the business decision process” (IMOS, 1997)
- “DS is utilising computer-based systems that facilitate the use of data, models, and structural decision processes in decision making” (Srivastava, 2001)

The need for decision support arises when a high degree of complexity is present in the decision problem. This complexity usually originates from (Bohanec & Rajkovic, 1990):

- A large number of parameters influencing the decision
- Incomplete knowledge
- Uncertain or conflicting goals
- Numerous and/or loosely defined options
- Different decision making groups with different objectives
- Time constraints imposed upon the decision making process

The above list alludes to the importance of information availability by including “incomplete knowledge” as a cause of difficulty. Morrison and Moore (1999) also relate decision support to the gathering of information. It is thus reasonable to conclude that a tool which generates information can be used effectively in support of decision making.

### 2.2.3. Decision support tools as part of decision support

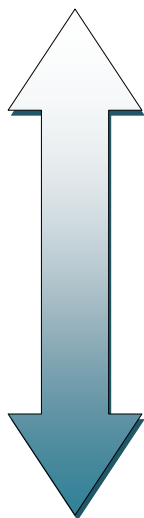
Decision support is made up of a number of more specialised disciplines, including operations research, decision analysis, and decision support tools/systems. Operations research is concerned with optimal decision making in systems that originate from real life, usually through modelling (Hillier & Lieberman, 2000), while decision analysis is defined by Keeney (1982) as “a formalisation of common sense for decision problems which are too complex for informal use of common sense.” Decision analysis usually proceeds by building models and using them to perform various analyses and simulations such as “what-if” and sensitivity analysis (Clemen, 1996).

“Decision support systems” (DSS) is the area of the information systems discipline that is focused on making decision making more effective (Arnott, 2005). A DSS is an interactive computer-based system (Sprague & Carlson, 1982; Lui, 2009) which supports managerial activities in decisions that are semi-structured (Ken & Morton, 1978). The information provided by DSS is distinguished from periodical reports in the way that the user accesses the information. The DSS user often initiates each instance of system use (Alter, 1977).

### 2.2.4. Classification of decision support systems

According to Alter (1977), DSS can be categorized according to the degree to which the system’s outputs could directly determine the decision. This is related to generic operations that can be performed by a DSS, ranging from purely data orientated to purely model orientated.

#### DATA-DRIVEN



#### MODEL-DRIVEN

- Retrieving a single item of information
- Providing a tool for ad hoc data analysis
- Providing representations of data in the form of reports
- Estimating the outcomes of proposed decisions
- Proposing decisions
- Making decisions



Alter (1977) argues that a DSS can be categorized by what type of generic operation it performs.

- **File drawer systems** provide access to particular data items. The hands-on users of these systems are typically non-managerial personnel, who use the system to support their day-to-day tasks. The system provides people who perform ongoing operational tasks with immediate access to the information they need.
- **Data analysis systems** are generally used by non-managerial staff to analyse files of current or historical data. Some data analysis systems provide the user with the capability to analyse data by means of summarisations, calculations, and pictorial representation. Microsoft Office Excel is an example of such a system.
- **Analysis information systems** aim to provide management information through the use of a series of decision-orientated databases and small models.
- **Accounting models** use definitional relationships and formulas to calculate the outcome of particular actions. One of the key attributes of these systems is managers' ability to understand them easily.
- **Representational models** include all simulation models which are not accounting based, estimating the consequences of actions, environmental conditions, or relationships. According to the author these models are mostly aimed at assisting managers in planning activities. Representational models attempt to develop an understanding of how future actions are related to future outcomes. One of the main issues is whether the model is a reasonable representation of the real-life situation that is studied. Representational models also tend to have credibility problems, because it is often possible to question the approximations used in modelling input-output relationships.
- **Optimisation models** are used to study situations whose goals involve combining parameters in such a way that attains a specific objective. An example of this type of analysis tool is linear programming.
- **Suggestion models** generate suggested courses of action based on formulas or mathematical procedures. The output generated is a definitive answer, rather than an analysis of tradeoffs and constraints.

Power (2001) consolidated Alter's classification system into three broader types of DSS, according to the dominant technology component or driver of the system. Alter's first three DSS types (file drawer systems, data analysis systems, analysis information systems) are called data-driven; the second three types (accounting models, representational models, optimisation models) are model-driven; and the last (suggestion models) has been called intelligent or knowledge-driven.

- **Data-driven DSS** facilitate access to and manipulation of large databases of structured data.
- **Model-driven DSS** emphasise access to and manipulation of statistical, financial, optimisation, or simulation models. They use data and parameters input by the user to analyse a situation and provide decision support.
- **Knowledge-driven DSS** provide problem-solving expertise in the form of facts, rules, or procedures. These can also be called Management Expert Systems, and are able to suggest or recommend actions to managers.

Contemporary DSS can be divided into seven groups (Arnott, 2005), although any given system can be classified as belonging to more than one group.

- **Personal decision support systems (PDSS).** These are small scale systems that are normally developed for either one manager or a small number of independent managers, which seeks to aid in one specific decision task. Modern PDSS can source data from data warehouses and employ powerful modelling approaches from management science and operations research.
- **Group support systems (GSS).** In the applications that GSS is designed for, a group of managers share the responsibility for a decision, and all are involved in the decision process. These systems aim to support a group of people engaged in a decision-related meeting (Huber, 1984).
- **Negotiation support systems (NSS).** Negotiation support systems are also designed for use in groups, but aim to facilitate negotiations by using computer technologies.
- **Intelligent decision support systems (IDSS).** Two types of IDSS exist: the first involves the use of rule-based expert systems, while the second makes use of neural networks, genetic algorithms and fuzzy logic (Turban et al, 2005).
- **Executive information systems (EIS) / Business Intelligence (BI).** Executive information systems are data driven systems that provide information about the state of the business to management (Fitzgerald, 1992). Business intelligence is a contemporary term for EIS, encompassing both data-driven and model-driven DSS that focus on management reporting.
- **Data warehouses.** Data warehouses provide large organisations with an integrated view of their business. They usually consist of a set of databases created to provide information to decision makers (Cooper et al, 2000). Data warehouses can also be used in conjunction with other DSS, by providing the raw data behind PDSS and EIS.

- **Knowledge Management-based DSS (KM).** Decision-making within an organisation can be supported by the management of what they deem as knowledge. A KM-based DSS aids knowledge storage, retrieval, transfer, and application.

The three classification systems discussed in this section can be used to guide the design of a decision support tool by incorporating application-specific criteria.

## 2.3. Discussion of blending literature

### 2.3.1. The function of a stockpile

Four functions of stockpiles are identified by De Wet (1983):

- Bridging interruptions in parts of the system without stopping the whole system
- Acting as a buffer between continuous and batch operations
- Collecting, storing, and distributing material coming from or going into different flow lines
- Homogenising, blending, and proportioning raw materials in order to prepare it for a downstream metallurgical or chemical process

A stockpile is seen to be successful in the last of these four functions if the instantaneous analysis of reclaimed material closely resembles the average value of the whole pile. This, according to De Wet, will depend on the storage capacity of the bed, the nature of variation in the input material, and the degree of quality control exercised.

### 2.3.2. Advantages and disadvantages of different stockpiling systems

Bond et al explore the advantages and disadvantages of different stockpiling systems, and note that the coneshell stacking method can be applied if the application does not require much pre-blending (Bond, Coursaux & Worthington, 2000). The authors do not however recommend that this stacking method be used where a high degree of variability is present in the raw material feed. The use of a full-face bridge reclaimer is recommended, which, in combination with chevron, windrow, or chevcon stacking, can typically achieve a 10:1 reduction in variation of the material chemistry. This result is valid provided that more than 300 layers are stacked and that the blending volume is large enough.

Furthermore Bond et al demonstrate the end-cone effect by analysing the LSF (lime saturation factor) of material reclaimed from a longitudinal stockpile. At both ends of the pile a 10% deviation from the otherwise uniform signal is observed. This is seen as a major reason for the move towards circular blending piles, but a disadvantage of circular piles is also highlighted: The useful storage volume of a circular pile depends on the distance between the machines. If the blending section's volume is increased, the total storage capacity of the pile is decreased. Therefore, relative to linear piles, the blending section is much smaller, resulting in a greater output variation.

In general, when chevron stacking is combined with either bridge scraper reclaimers or drum reclaimers a good blending ratio can be obtained. Similar results can be expected from circular stockpiles with chevron stacking and a bridge scraper reclaimer (Erasmus, 2001).

### 2.3.3. Bed blending research in South Africa

De Wet (1983) describes applications of blending technology in South Africa by making reference to the objectives and design criteria of homogenisation plants. The author furthermore discusses the different equipment types available, and details the decision criteria that a plant engineer would need to consider when designing a homogenisation facility.

### 2.3.4. Sampling theory

Many of the publications on bed-blending reference the work of P.M. Gy, a leading authority on sampling theory. His research spanned half a century, producing multiple books and articles. Sampling theory is relevant to stockpile simulation because the reclaiming phase can be seen as a systematic sampling of the input material (Gy, 1981). Another reason for including Gy's sampling theory in the study of blending piles is the introduction of a Fundamental Error and Heterogeneity Invariant (Gy, 1992). It sets a bound on the best possible blending that can be achieved without reducing the size of particles. According to Robinson (2004) the variance induced by sampling can be calculated as shown below:

$$V_s = \frac{(1-p)}{p} \sum_i \left[ \left( \frac{m_i}{M} \right)^2 (g_i - G)^2 \right]$$

Equation 12: Calculation of sampling error variance (Robinson, 2004)

In Equation 12,  $p$  is the probability that any given particle will be selected,  $m_i$  and  $g_i$  are the particle mass and grade respectively, and  $M$  and  $G$  are the overall mass and grade of the lot from which the sample is being taken. This represents a component of variation that cannot be eliminated, and should thus be added to any predicted value of output variation.

Denny and Harper (1962) also used sampling as a base for blending theory, saying that since reclaimers remove small cross-sectional increments of piles they perform a function analogous to sampling the whole pile. The authors likened the number of samples to the number of layers, and the sample frequency to the pace of the reclaimer's progression.

### 2.3.5. Efficiency of blending

Bed blending efficiency can be calculated by making use of statistical methods, but two important assumptions need to be made (Denny & Harper, 1962). First it must be assumed that the variable in question follows a normal distribution. Secondly it is assumed that the stacker moves at a speed that is sufficient to distribute all elements of the blend along the entire length of the pile.

According to Bond et al (2000) blending efficiency can be increased by increasing the volume of the stockpile or increasing the stacking speed (i.e. stacking more layers). The minimum number of stacked layers required to ensure proper blending:

$$N = \frac{T \times A \times D \times 60}{C}$$

**Equation 13: Minimum number of stacked layers for proper blending (SACPS, 2011)**

In Equation 13,  $N$  is the number of layers,  $T$  is the travel speed of the reclaimer in meters per minute,  $A$  is the cross-sectional area of the stockpile in square meters,  $D$  is the bulk density of the material in tonnes per cubic meter, and  $C$  is the capacity of the stacker in tonnes per hour. Variations in thickness throughout a layer are generally of little importance, provided that the pile is of adequate length and more than 200 layers are stacked (Mikkelsen, 1983).

De Wet (1994) stated that, subject to certain statistical assumptions, the homogenising effect of a blending pile can be estimated as:

$$\frac{S_{in}}{S_{out}} = 0.5 \sqrt{N} = \frac{60 \times V \times F}{Q}$$

**Equation 14: Estimation of the homogenising effect of a blending pile (De Wet, 1994)**

In Equation 12  $S$  is standard deviation and  $N$  is the number of layers the reclaimer cuts into. In the second part of the equation  $V$  is the stacker travel speed in m/min,  $F$  is the stockpile cross-sectional area in  $m^2$ , and  $Q$  is the stacking rate in  $m^3/h$ . Denny and Harper (1962) estimated that the standard deviation of the resulting blend from a linear stockpile would be decreased by a factor of  $\frac{1}{\sqrt{N}}$ , where  $N$  is the number of layers intersected by the reclaimer. This estimation was echoed by Petersen (2004), ignoring the effect of sampling and analytical errors.

### 2.3.6. Modelling of blending piles

This section will discuss some examples found in literature of modelling and simulation work that will be applicable to this study.

Gerstel (1979) developed statistical models to predict the autocorrelation function (the pattern formed by variations in grade) of outputs from several different types of blending piles (Gerstel, 1979). In further work Gerstel developed a mathematical model for circular blending piles to offer a reasonable estimation of the homogenisation that can be obtained (Gerstel, 1989). The author noted that in this estimation varying mass of the input flow is not considered.

Hurwitz and Ackermann registered a patent called “Real-time optimisation for mix beds”. It describes a computerised stockpile management tool that uses information about the stacker system as inputs (stacker

location, material delivery rate, material composition, mass flow rate to stacker, desired aggregate composition) to output a position for the stacker to move to (Hurwitz & Ackermann, 1999). It creates a real time database of the aggregate composition of materials in respective parts of the pile, and predicts the composition of these parts when they are reclaimed. The stockpile profile is modelled in terms of vertical sectors, and the stacker is moved forward and backward to stack incoming material on specified sectors and (ideally) create a pile of perfectly blended composition.

Bond et al (2000) hypothesise that if real-time data obtained from online analysers were combined with the appropriate control software it could be used to modify the stacking location according to the incoming material. This would be done by varying the stacker speed to place material in a way that would improve the overall homogeneity and average grade of material on the stockpile. They stated that this kind of system would have the greatest impact on circular stockpiles, as their blending sections have smaller volumes. By performing simulations the authors proved that this “optimised stacking” resulted in less variation than constant speed stacking.

Robinson (2004) uses statistical and geometrical modelling to predict the variation that can be expected from different linear stockpile configurations. The author starts by developing a simple bed-blending model where a reclaim slice contains an equal proportion of each layer of material stacked. This can be visualised as placing material into a two-dimensional array of containers. Variation in material flow is ignored, which leads to the assumption of equal masses being deposited into each of the containers. Containers are filled row-by-row and reclaimed column-by-column. By example: If material is stacked in 1000t blocks from number 1 to 20 as shown in Table 1. The five reclaimer slices would then contain equal proportions of blocks {1 10 11 20}, {2 9 12 19}, {3 8 13 18}, {4 7 14 17}, and {5 6 15 16}. Resultantly statistical expressions can be derived, depending on the variation of properties in the input stream, to predict the variation in the reclaimed material. This model would only provide an adequate approximation for the middle of some piles with simple geometry.

**Table 1: Representation of bed-blending model**

20	19	18	17	16
11	12	13	14	15
10	9	8	7	6
1	2	3	4	5

Robinson extended his modelling by taking into account more realistic geometries, where the reclaim slices will have unequal proportions of the layers of stacked material. As was also noted by Petersen (2004), all layers will not contribute the same amount of material to each reclaimed slice due to the inclined nature of the reclaimer face relative to the stacked layers. By using the principle above combined with modelling of layer geometry Robinson generated a matrix with values of  $m_{i,j}$ , i.e. the mass of layer  $i$  that is reclaimed in slice  $j$ . The mass matrix can be normalised to generate a matrix of proportions, which in turn can be used to calculate the weighted average grade of the reclaimed slices if the grades of stacked layers are known. The model developed is used to compare the expected output variation from different combinations of stacking and reclaiming methods, assuming that the grade of material varies slowly compared to the length of a stacker pass. Two stacking methods, namely windrow and chevron, and two reclaiming methods, bench and full-width, were used. It was found that a chevron-stacked, full-width-reclaimed linear stockpile would produce the best homogenisation performance, with a windrow-stacked, bench-reclaimed pile delivering the least reduction in variation.

Robinson's work was used by Kumral (2006) to build a stockpile simulator. Kumral models a stockpile as a set of blocks created by the intersection of reclaimed slices and stacked layers. He then simulates grades for each of these blocks. The simulator is used to construct a multiple regression model with number of blocks in each layer, stacker speed, and stacking type as the independent variables, VRR as the dependent variable. Finally genetic algorithms are used to find optimal stockpile parameters for a minimal VRR. The method is applied to a case study where the stockpile input was simulated at 30000 locations, and it is concluded that this approach can be used to design mineral stockpiles. Kumral also uses the simulation to compare windrow and chevron stacking methods, and concludes that windrow stacking delivers a lower VRR. Robinson has published his comments on Kumral's work, disputing Kumral's conclusion on windrow stacking, as it "is not consistent with his own model" (Kumral, 2006). Robinson uses an example to show that Kumral's model delivers the same predictions of blending performance for both windrow and chevron stacking. Robinson's point is supported by an earlier statement by Mikkelsen (1983), who said that there is no practical difference in blending between windrow and chevron methods.

In Kumral's earlier work he also addressed the blending optimisation problem, by predicting the optimal blending ratios of different ores needed to produce the required blended ore (Kumral, 2005). The optimisation model was developed by using quadratic programming.

Pavloudakis and Agioutantis (2003) developed a simulation model using the Visual Basic for Applications platform by Microsoft. The simulation uses input describing the properties of material that is stacked, and combines it with information about the stockpile geometry to output a series of bulk solid property values. The research was focussed on linear stockpiles only, assuming constant stacking and reclaiming rates. The effects of segregation and variation at the end-cones were also ignored.

### 2.3.7. Decision support tools in stockpile management and quality control

Examples of decision support tools applied to stockpile and quality management can be readily found in literature, some of which are discussed below.

An example of a knowledge based DSS can be seen in the work of Micali and Heunis from Eskom, which they called the Coal Stock Pile Simulator (CSPS) (Micali & Heunis, 2011). In order for Eskom to safeguard the delivery of energy from coal fired power stations, each station needs to have a buffer capacity of coal reserves in stock. Careful management of these stockpiles is of utmost importance. The CSPS is a simulation model providing “what-if” scenarios and planning sets for decision making. The CSPS is not an optimisation model, meaning that it can evaluate a plan, but it cannot produce a plan or choose an optimum plan. It can however be used to optimise situations with the use of heuristics. Monte Carlo simulation was used to generate possible stockpile level scenarios, and statistical models were developed using first order multiple regression to represent uncertainty in the system. The application was developed using Microsoft Excel as a platform, and was iteratively improved during implementation.

Everett (2007) developed a decision support tool for use in the iron ore industry in Western Australia. The tool would aid mine operators in deciding which blocks of ore to mine each day, with the end goal of achieving a uniform grade of acceptable quality. It makes use of exponentially smoothed grade data of mined ore to calculate the grade vector of the ore yet to be mined for a specific day. The tool was written by using Visual Basic macros for Microsoft Excel, so that it could be easily accessible to operators and planners. Everett emphasised that the success of this tool is dependent on the operator/planner using it as a supplement to their knowledge of the domain, not as a replacement. The developed tool would be a decision *support* system, not a decision-making system – a “fruitful example of human/computer interaction” (Everett & Kamperman, 1999). The final product of the research was the Continuous Stockpile Management System (CSMS), which comprises of an Excel workbook with multiple sheets to receive input and execute the macros. In the conclusion of his paper Everett says that this approach was preferred over using specialised computer packages, because it becomes easier to involve operational, planning, and management staff. The CSMS developed by Everett was implemented at the BHP Billiton Iron Ore Yandi Fines Operation, providing significant improvements in resource utilisation, operational efficiencies and costs (Kamperman, Howard & Everett, 2003). This followed successful implementation at Mount Newman, also a BHP Billiton Iron Ore operation (Everett & Kamperman, 1999).

Petersen (2004) modelled the expected variation in output parameters of a cement blending pile by using statistical methods and sampling theory. The author noted that it cannot be expected from the process operator managing the mixing pile to have in-depth knowledge of sampling and statistics (Petersen, 2004), and he therefore converted the statistical calculations to a Microsoft Excel based decision support tool. By



entering basic data into the worksheets, operators can estimate the standard deviation of reclaimed material and investigate the influence of changes in the feed sequence on blending efficiency.

Gupta et al (2007) developed a blending model for Indian coking coals. It uses relationships between coke quality parameters and coal quality parameters to suggest a least-cost blend of coal to achieve targeted coke quality. The model was validated with known data from the Steel Authority of India Limited (SAIL), and found to give “quite accurate” predictions of coke quality for a given coal blend.

## 2.4. Applications of bed-blending research

This section will highlight the relevance of bed-blending research to three key commodities.

### 2.4.1. Applications in energy coal

As the world’s high quality coal reserves are being depleted mines are forced to beneficiate coal from seams of lower quality and consistency. Wider variations in grade can therefore be expected. Even when the average grade of a consignment of coal is of the right standard, individual increments may differ widely and make utilisation difficult. Bed-blending has been applied successfully in preparation of coal in Britain and India for years (Denny & Harper, 1962).

Coal processing relies on gravity separation to beneficiate a target grade of product coal. A mixture of water and magnetite is prepared to “wash” the coal feed at a specified density. Lighter coal particles are of better quality, and will separate from the heavier particles in one of a variety of processing equipment. The density of the washing medium is controlled according to the grade fed to the plant and the desired output grade. Frequent adjustments to this density, to account for high variability in feed material, are not practical. De Wet (1994) uses this argument to motivate that it would be impossible to produce consistent product from a coal washing plant without making use of pre-homogenisation. Also, many coal properties are not related to specific gravity, and can therefore not be regulated by the coal washing process. Consistency in these properties can only be obtained by homogenising the feed (Denny & Harper, 1962). The more uniform the properties of the raw coal feed, the cheaper and more efficient the separation process (Lemke, 1962).

### 2.4.2. Applications in cement

A large proportion of research on increasing blending efficiency has been focussed on the cement industry. This is because of the high sensitivity of the cement process to any fluctuations in input grade. Pre-blending is mostly applied to the main constituent of cement raw material, namely limestone (Mikkelsen, 1983).

The most common type of cement used worldwide is Portland cement (Carpio, Coelho, Silva & Jorge, 2005). The process can be summarised as shown in Figure 13. Clinker is produced in a rotary kiln, fired by coking coal. Pre-blending and homogenisation are useful in preparation of the clinker raw materials and to ensure

constant coke composition. Coke plays two important roles: generating or reducing gas and heat. Therefore consistent coke quality is one of the significant factors driving blast furnace performance (Gupta, Das & Chauhan, 2007).

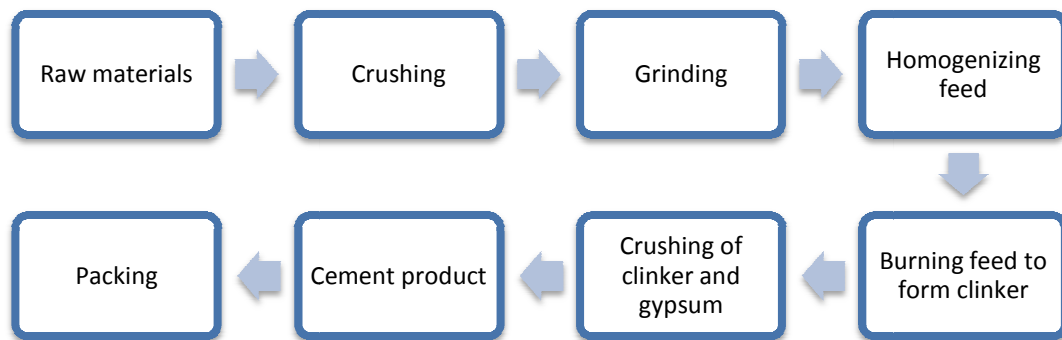


Figure 13: Process steps in the manufacture of Portland cement – Adapted from (Carpio, Coelho, Silva & Jorge, 2005)

Bond et al (2000) discuss advances in stockpiling systems, sensors for monitoring raw-material chemistry, and new designs for homogenisation silos, all in light of producing a uniform feed to the cement manufacturing process.

#### 2.4.3. Applications in iron ore

Iron ore is mainly used in the steel industry. Steel makers feed the ore into blast furnaces, which are finely tuned to certain ore specifications. Therefore the delivered ore must be consistently close to target grade. According to Everett (2007), the target composition is typically of the order as shown in Figure 14. The consistency of this composition in the delivered ore is the main quality criterion, an important factor in price and quantity negotiations (Everett, 2007).

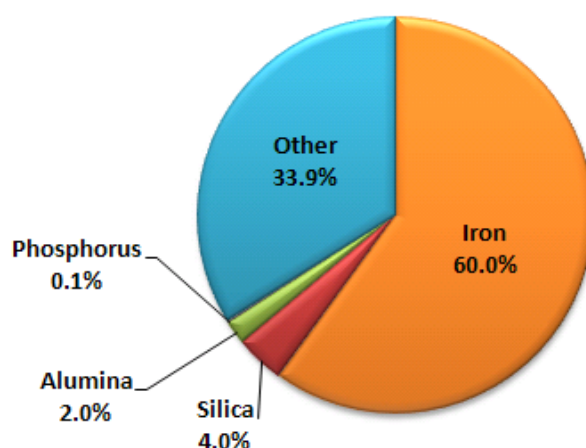


Figure 14: Target composition of iron ore for steel production

Bed-blending was first used in 1938 to reduce the variation of iron and alumina content in British iron ores (Denny & Harper, 1962). Substantial benefits were achieved, with the coke rates of British blast furnaces falling by nearly 10 per cent. Before introduction of pre-homogenisation burden charges had to be made up once in every 8-hour shift, where furnaces were now able to operate at constant burden make-ups for periods of 30 days. The blending process not only regulated the iron content of the ore, but also the concentration of lime and silica.

## 2.5. Summary of blending literature discussion

From the works of Bond et al and Erasmus it was concluded that if high homogenisation efficiency is desired the best stockpiling system would be one that uses chevron or chevcon stacking combined with full-face reclaiming. Robinson also came to this conclusion, by using his bed-blending model to compare chevcon and windrow stacking methods. Furthermore he found that homogenisation efficiency is at its worst when windrow stacking is combined with bench reclaiming. Coneshell stacking was deemed adequate only when blending is not a primary design consideration. A potential downfall of circular piles was identified by Bond et al. The blending section of a circular pile is much smaller than for linear piles, since an increase in the blending volume would decrease the overall storage capacity of the pile.

Robinson used sampling theory developed by Gy to demonstrate a way in which the minimum variation in the output of a blending operation can be calculated. Sampling theory formed the base of bed-blending research, since the way that reclaimers remove small cross-sectional increments of piles can be seen as analogous to sampling of the whole pile. The number of samples can be likened to the number of layers the reclaimer blends into each slice, and the sample frequency the pace of the reclaimer's progression (Denny & Harper, 1962).

Most of the work done on calculating blending efficiency uses statistical methods, subject to some assumptions. These include a normal distribution of the raw material, and that the stacker moves at a speed that is sufficient to deposit material uniformly along the length of the blending volume. Blending efficiency can be increased by stacking more layers or increasing the stockpile volume. Denny and Harper, De Wet, and Petersen all presented estimations of blending efficiency based on the number of layers stacked.

Gerstel developed a mathematical model, based on statistical approximation, to offer a reasonable estimation of the homogenisation that can be obtained by using a circular blending pile if constant stacker feed is assumed. Hurwitz and Ackermann registered a patent for their bed-blending optimisation model, which created a database of the grades of materials in the pile and predicted the aggregate output composition of vertical sectors when they are reclaimed. Important stockpile modelling work was done by Robinson, who used statistical and geometrical modelling to predict the expected variation in output

properties from different linear stockpile configurations. The model developed used a matrix of proportions representing the amount of material from a layer that is contained in an inclined material slice, again assuming constant stacker feed. Kumral used Robinson's modelling principle to build a stockpile simulator, modelling a stockpile as a set of blocks created by the intersection of reclaimed slices and stacked layers. The simulator is used to build a multiple regression model for optimising stockpile parameters, by minimising VRR for different scenarios. Pavloudakis and Agioutantis developed a simulation model using the VBA platform that uses properties of stacked material combined with the stockpile geometry to predict the output from a linear stockpile. The end-cone sections were ignored, modelling only the triangular prism-shaped part of the pile.

Examples of DSS used in stockpile simulation and quality control applications were reviewed. Micali and Heunis developed a knowledge based DSS for Eskom to manage their stockpiles by providing "what-if" scenarios and planning sets for decision-making. A decision support tool developed by Everett was implemented in multiple iron ore operations in Western Australia, providing significant improvements in resource utilisation, operational efficiencies, and costs. The model used the VBA platform and served to aid mine operators in deciding which blocks of ore to mine each day in order to achieve a uniform grade of acceptable quality. Everett emphasized the importance of the user as part of the decision-making process. Petersen modelled the output from a cement blending pile using statistical methods and sampling theory, but acknowledged that most operators who make decisions about the management of these piles do not have in-depth knowledge of sampling and statistics. A Microsoft Excel based decision support tool was therefore developed so operators could use Petersen's methods to investigate the influence of changes to the stockpile system.

### 3. Research design

#### 3.1. Research approach

As discussed in section 2.2, decision support systems present relevant information to the user thereof. By having required information available the user can make decisions more easily. The decision support tool envisioned as output to this research project would provide information about the expected grade variation when material is reclaimed from a circular blending pile.

Before any DSS can be designed it is important to identify the most crucial requirements that the system would need to comply with. Within the context of stockpile management, and the types of questions that the decision support tool would aim to answer, the following criteria were identified:

- ✓ The DSS needs to provide information about a complex real-life system
- ✓ The DSS needs to be able to predict the consequences of decisions and actions
- ✓ Decisions will mostly be made by individuals, i.e. non-collaborative decision making
- ✓ The users of the system would be production managers and production planners

In Table 2 different DSS types, as classified by three authors, are evaluated at the hand of the criteria stated above. When Power's classification is used, a model-driven DSS is necessary. Alter's classification narrows this description to specify that the planned DSS needs to make use of a representational model. According to Arnott's classification the newly developed DSS will be part of the PDSS family. The fact that the DSS will be driven by information from a model and focussed on management reporting means that it can also be classified as part of business intelligence.

A model is a computer representation of reality that allows the user to investigate the impact of a possible decision (Liu, 2009). A model-driven DSS can be seen as consisting of four parts, summarised in Figure 15 (Power, 2002). According to Shim et al (2002) a DSS consists of three basic capability components: a data management component, a model management component, and a communication management component. These are represented by the database, model, and user interface respectively. The model gets input from the user through the user-interface, and from the database. It processes the input and delivers feedback to the user, again through the user-interface. It is important to note that the user forms an integral part of any DSS (Marakas, 1999; Power, 1999).

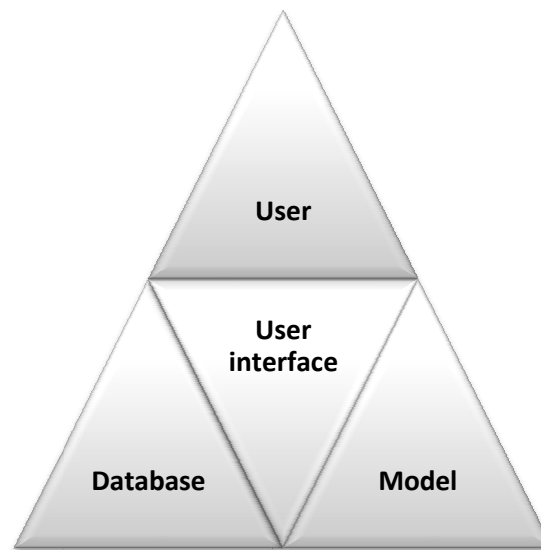


Figure 15: Components of a model-driven DSS

The tool was developed by using Visual Basic for Applications (VBA). VBA is a product from Microsoft that is released as a standard part of their Office package (Walkenbach, 2007). Procedures called macros are written in VBA and used to execute commands in any of the Microsoft Office programs. The VBA platform was chosen for development of the decision support tool after consideration of the following:

- ✓ VBA interacts seamlessly with Microsoft Office Excel, which is used widely for data entry and storage
- ✓ It requires no additional software to be installed on control room computers and/or the personal computers of managers and other decision makers
- ✓ A user interface can be written which is easy to navigate and simple to understand. It enables users to access the complex methods used in the simulation model development without needing to understand how they work
- ✓ The VBA model will deliver consistent results. Since all calculations happen in the background, human error can be minimised
- ✓ Minimum training will be required as potential users will all have moderate proficiency in Microsoft Office Excel

VBA was used to develop two simulation models, one each for circular stockpiles stacked in coneshell and chevcon modes respectively. Sheets are provided for data entry, and a central user interface controls the simulation parameters. The models were developed from first principles, and subsequently tested for accuracy by comparing modelled versus recorded data from a case study. Finally the influence of different stockpile parameters on the blending efficiency achieved is investigated by means of a sensitivity analysis

Table 2: Identification of suitable DSS types

	Provide information about a complex real-life system	Predict consequences of actions/decisions	Individual users (non-collaborative decision making)	Users = production planning and production management
<b>Power (2001)</b>				
Data-driven DSS			x	
Model-driven DSS	x	x	x	x
Knowledge-driven DSS		x	x	x
<b>Alter (1977)</b>				
File drawer systems			x	
Data analysis systems			x	
Analysis information systems			x	x
Accounting models		x	x	x
Representational models	x	x	x	x
Optimisation models		x	x	x
Suggestion models		x	x	x
<b>Arnott (2005)</b>				
Personal DSS	x	x	x	x
Group support systems				x
Negotiation support systems				x
Intelligent DSS		x	x	x
EIS / Business intelligence	x	x	x	x
Data warehousing			x	x
Knowledge Management-based DSS			x	x

### 3.2. Development of a simulation model

Simulation is a management science technique that can be used to analyse and study complex systems (Winston, 2004). It imitates the evolution of a real-world system over time, usually by developing a model. A circular blending pile will be simulated as a deterministic continuous system which changes dynamically. No statistical assumptions or probabilities are included, and variables will change continuously instead of as a result of discrete events.

The principle behind the model design is that the entire stockpile volume can be envisioned as being made up of many small blocks. This echoes the ideas of Robinson (2004) and Kumral (2006), but is approached in an entirely different way. Instead of stacking individual blocks of set volume, and reclaiming them in another order to achieve a blend, a Cartesian coordinate system is used to model the placement and properties of each block.

A three-dimensional grid can be drawn; large enough for the entire stockpile to fit inside (see Figure 16). The grid corresponds to three Cartesian axes, with the x- and y-axis forming the base of the stockpile and material being stacked upwards in the increasing z-direction. Therefore every block in the grid has an address, which can be represented by using the position of the block in the Cartesian coordinate system.

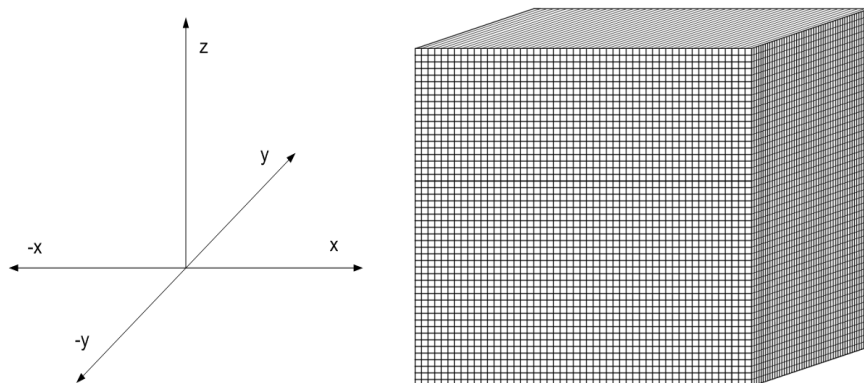


Figure 16: Block model on Cartesian axes

The grid is divided so that blocks have side lengths of one meter, which means that each block has a volume of one cubic meter. This is equal to roughly one tonne of material, depending on the material type and bulk density. In light of feed rates to stockpiles in the order of a thousand tonnes per hour it is thus reasonable to assume minimal variation of grade within each of these blocks. When material is stacked the blocks fill up according to the stockpile geometry. In the way that this is executed there are two important things to note, best demonstrated by use of a simplified example. Figure 17 is a two-dimensional representation of what happens when blocks are filled within the grid. Two points to note are that: a) there are many blocks that are not filled at all, and b) some blocks are filled only partially. Therefore the simulation model needs to calculate the exact stockpile height at every point in the grid to build a more accurate grade profile.



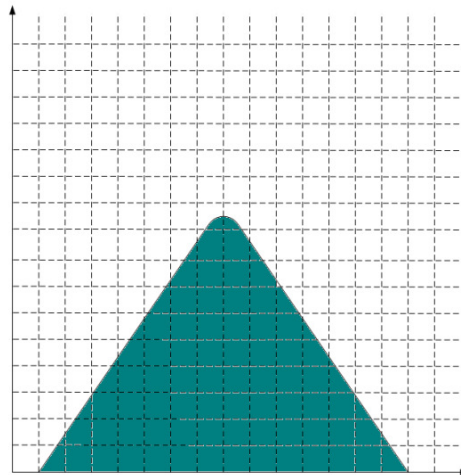


Figure 17: Two-dimensional representation of how blocks are filled

### 3.3. Model validation

In order to prove the model's validity it is compared to known outputs. For this purpose data was collected from a South African coal processing facility. The simulation model was used to predict the behaviour of a stockpile, which was compared to the actual behaviour recorded. The case study facility, dubbed Colliery X, uses a circular product stockpile, stacked from the coal upgrading process and reclaimed to send to the customer. The customer has strict specifications on product grade, and low variability is rewarded with financial incentives. The stockpile has a live capacity of 45kT, and material is sampled before it is stacked and after it is reclaimed, on 2 hour intervals. A full-face harrow reclaimer is used to scrape material from the pile cross-section, and the stacker currently operates in coneshell mode only. An upgrade to chevcon stacking is planned.

### 3.4. Sensitivity analysis

Once it is confirmed that the simulation model offers a reasonable approximation of reality, it can be used to evaluate the influence of some operational parameters on the blending performance of the stockpile. This is a useful function of a simulation model; the effect of changes can be investigated without actual implementation, and potential negative outcomes can be avoided. Users can therefore optimise their process through risk-free experimentation.

For the coneshell model two variables were chosen for the purpose of investigation. The increment that the stacker moves between successive cones can be varied and could possibly influence the efficiency of blending. At Colliery X it was observed that the stacker boom is not always raised to its maximum height during stacking. The influence of this practice on homogenisation was also investigated. Table 3 shows the list of coneshell simulations performed to generate the results discussed in section 6.1.

The influences of three variables were investigated for the chevcon model. This was the length of the blending tail, i.e. the angle of rotation, the stacker movement speed, and the number of incremental layers contacted by the reclaimer when cutting into the stockpile cross-section. When the number of layers is increased the increment that the stacker moves between steps is decreased. Therefore varying this parameter can also be seen as varying the stacker step increment. Details of the experimental simulations performed on the chevcon model are shown in Table 4.

Once all simulations are complete the results are compared on the basis of blending efficiency (VRR), in order to assess the influence of each variable. The results obtained for chevcon and coneshell methods are also compared to each other, in order to quantify the difference in potential blending efficiency achieved by using either method.

**Table 3: Simulations performed - coneshell model**

Coneshell stacking		
	Stacker increment (degrees)	Stacked height (m)
Simulation 1 (base case)	10	17
Simulation 2	6	17
Simulation 3	8	17
Simulation 4	12	17
Simulation 5	14	17
Simulation 6	10	15
Simulation 7	10	16
Simulation 8	10	18
Simulation 9	10	19

Table 4: Simulations performed - chevcon model

Chevcon stacking			
	Angle of rotation (degrees)	Stacker speed (m/min)	Layers stacked
Simulation 1 (base case)	100	25	200
Simulation 2	50	25	200
Simulation 3	75	25	200
Simulation 4	125	25	200
Simulation 5	150	25	200
Simulation 6	100	20	200
Simulation 7	100	22.5	200
Simulation 8	100	27.5	200
Simulation 9	100	30	200
Simulation 10	100	25	100
Simulation 11	100	25	150
Simulation 12	100	25	250
Simulation 13	100	25	300

## 4. Simulation model development

This chapter documents in detail the steps followed in developing the stockpile simulation model, including discussion of the VBA code. It starts with a discussion of the overall geometric modelling procedures used for both the coneshell and chevcon methods, followed by the specific methods used to execute variable speed stacking. Thereafter the reclaimer model is detailed, after which the chapter is concluded with a presentation of the user interface that was developed.

### 4.1. Geometric modelling

As stated in the first objective (section 1.2), development of the simulation model had to start with geometric modelling. The bulk of work that is required in predicting the output from a blending stockpile is in modelling complex stockpile geometries (Robinson, 2004). Two approximations were used in the geometric model development. It was assumed that the cross-section of the pile, as seen by the reclaimer, is triangular in shape. It was also assumed that the blending tail formed by the chevcon method, when cut on the line of the stacker path, forms a cross-section resembling a triangle. Trigonometric relations can therefore be used to relate the height of the tail at any point to its distance from the tail end.

#### 4.1.1. Coneshell stacking model

Model building started with the coneshell method, and a Microsoft Excel workbook named “Coneshell stacking.xlsm”. For the objective of geometric modelling two worksheets were created:

- “GUI” – A preliminary graphic user interface which would be used during the development process to execute all VBA macros and input parameters relative to the execution.
- “GeoData” – A matrix of cells, the size determined by the diameter of the stockpile. Each cell corresponds to x and y coordinates related to its column and row index respectively. The value stored in the cell is the maximum height of the stockpile at the specified point.

Input to the geometric model makes use of certain constant stockpile and material properties<sup>2</sup>, summarised in Table 5. The values indicated were obtained from the stockpile design used by Colliery X, but would be adjustable variables in the final user interface.

---

<sup>2</sup> These properties are defined in detail in section 2.1.3

Table 5: Constant properties used as input to the geometric model

Property type	Property	Variable Name	Value	Unit
Material	Angle of repose	<b>aRepose</b>	38	Degrees
	Bulk density	<b>bulkDens</b>	0.9	t / m <sup>3</sup>
Stockpile	Diameter <sup>3</sup>	<b>diam</b>	110	m
	Stacker radius	<b>rStack</b>	28	m
	Maximum height	<b>hStockpile</b>	19	m

All procedures for stacking material are called from a central **Stack**<sup>4</sup> macro, henceforth referred to as the stacker model. It starts by calling a macro named **defineEverything**, where values are assigned to all relevant variables by reading from the “GUI” worksheet. Any properties measured as an angle are input in degrees, as it is a unit that is more generally understood, but converted to radians in the definition step. This is because VBA uses radians as a default unit for measuring angles. Some properties are derived from others, for instance:

- **rCone**. This is the radius of a fully stacked cone, which can be found by a simple trigonometric relationship as:

$$rCone = \frac{hStockpile}{\tan(aRepose)}$$

Equation 15: Trigonometric relationship between the cone radius, angle of repose, and stockpile height

When a new stockpile is created, four arrays are defined:

- **CartArray (x,y,z)**. A three-dimensional array that is described by three indices, representative of a point's x-, y-, and z- coordinates in the Cartesian plane. The size of the array is determined by the diameter of the stockpile. Both x- and y- coordinates range from (-radius) to (radius), which means that the array would have (**diam**+1) locations in each direction. The range of z-coordinates will always be positive, and depend on the maximum height of the stockpile. This was assumed to never be more than a quarter of the stockpile diameter, which means the z-coordinates range from 0 to (**diam**/4). **CartArray** is used to store the grade values of each point on the stockpile. By example, if “**CartArray(10,20,5)** = 22.5” it means that the average grade of particles in the section located 10 meters in the positive x-direction, 20 meters in the positive y-direction, and 5 meters vertically upward is 22.5. **CartArray** is defined as a global variable, which means that it retains its value between executions of the code. This is important in order for grade data to be kept until the material is reclaimed from the pile.

<sup>3</sup> Measured on the reclaimer rail track

<sup>4</sup> The names of procedures and variables are indicated in **bold** typeface, in the interest of clarity

- **Radial (i).** A one-dimensional array representing all points on the base level of the stockpile. The array is thus comprised of  $(\text{diam} + 1) \times (\text{diam} + 1)$  elements. Each element in the **Radial** array stores the radial part of a point's polar coordinates (see section 2.1.6), and the points are numbered as demonstrated in Figure 18.
- **Angular (i).** Similar to the **Radial** array, except that it stores the angular part of the point's polar coordinates.
- **Heights (i).** Similar to **Radial** and **Angular**. For any given point on the base this array stores the height of the stockpile at that point.

**CartArray**, **Radial**, **Angular**, and **Heights** are all defined as global arrays, which means that they only need to be redefined if a brand new stockpile is created, i.e. if the model is applied to a different stockpile for the first time.

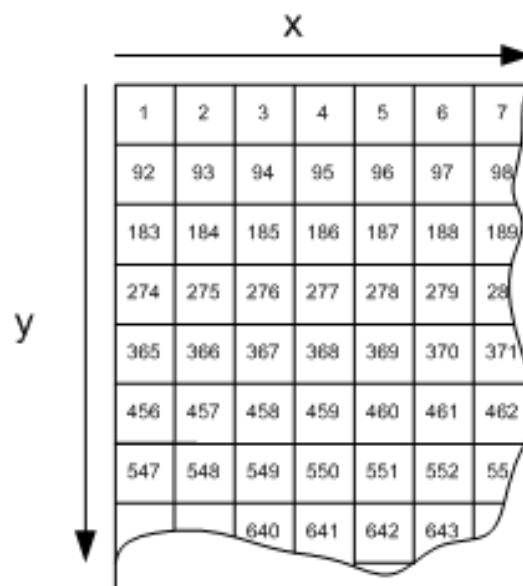


Figure 18: Counter numbers for Radial, Angular, and Height arrays

The next step in constructing the geometric model is to run a macro called **CreatePoints**. The purpose is to populate the **Radial** and **Angular** arrays by assigning polar coordinates to each of the points on the base level of the stockpile. Figure 18 shows how the procedure loops through coordinates and equates them to counter numbers. For example, **Radial(367)** would refer to the radial component of the polar coordinates for the point located at (3;5) in the Cartesian plane.

For each coordinate Equation 5 is used to calculate the radial component. Before the angular component can be found, one has to first determine in which Cartesian quadrant the point is located. This is done because the *arctan* function can return more than one solution, and in VBA the solution that lies in the first quadrant is returned as a default. The x- and y-coordinates are evaluated, as shown in Table 6, to

determine in which quadrant each point is located. The first quadrant answer returned by Equation 6 is then adjusted accordingly. **CreatePoints** also accounts for where Equation 6 is not defined, i.e. where  $x = 0$ , by adjusting the angle to either  $90^\circ$  or  $270^\circ$ , whichever is applicable.

Table 6: Quadrant adjustment in CreatePoints

x	y	Quadrant	Adjustment
>0	>0	1	angle (no adjustment)
<0	>0	2	$180^\circ - \text{angle}$
<0	<0	3	$180^\circ + \text{angle}$
>0	<0	4	$360^\circ - \text{angle}$

#### 4.1.2. Chevcon stacking model

A second workbook called “Chevcon stacking.xlsm” was created to model the behaviour of the stockpile when stacking in Chevcon mode. For the most part it is similar to the coneshell stacking method, but differences will be highlighted below.

The stockpile parameters that are input into the “GUI” sheet (shown in Table 5) remain the same since, in the interest of comparison, the same stockpile has to be modelled for both methods. These parameters are also read into the stacker model by using the **defineEverything** macro, with the four global arrays (**CartArray**, **Radial**, **Angular**, **Heights**) being defined as in the coneshell model. **CreatePoints** is again used to assign polar coordinates to all points in the Cartesian array.

The variable **rCone** is replaced by **rBase**. It is calculated in the same way as shown in Equation 15, but now refers to half of the fully stacked base width of the chevcon pile.

## 4.2. Stacking with variable feed rate

This section of the model development seeks to eliminate an assumption which is present in most of the literature reviewed; that of constant stacker feed. Two other assumptions remain relevant though. Firstly, it is assumed that the stacker speed is high enough that grade and flow variation within a layer is negligible. Therefore a layer is uniformly distributed in terms of thickness and grade. Secondly the effect of segregation, as shown in Figure 10, is not considered. This section also sees the introduction of two additional worksheets:

- “CalcSheet” – A sheet reserved for the execution of miscellaneous calculations, including the iterative determination of coneshell radii (section 4.2.2).
- “DecData” – A sheet containing data on feed rates and grades of material sent to the stacker

Before the simulation can start, the model needs to know whether it should stack a stockpile from “scratch”, i.e. start building a new first cone, or build onto the stockpile that is already in place. The “GUI” sheet features a radio button that is set to either “Start a new pile” or “Continuous stacking”. The radio button controls the value of the variable named **stackFromScratch**, and will mostly be set to “Continuous stacking” in normal operation. The following simulation parameters are also added to the “GUI” worksheet, and read into the model using the **defineEverything** macro:

- The stacker starting position (in degrees) : **aStack**
- The simulation duration – how long the virtual stacker should place material before stopping: **stackerSimTime**

At this point the first data is read into the stacker model. During development of the simulation model it was assumed that grade and volume data is available at regular increments, which would be the case during normal operation of a production facility. Information about the material fed to the stacker should be arranged in a data sheet with three columns. The first column is optional, and is used to store the timestamp of the data point. The stacker model does not use the actual time at which a data point was generated, only the interval between data points, so it can be omitted. This column will be useful when data points aren’t generated at precise intervals though, for instance with manual sampling<sup>5</sup>. The second column contains the tonnes of material fed in the particular time interval, and the third column contains the corresponding grade of material.

At the top of the data sheet two parameters are stored. The first is the time interval between consecutive data points, calculated by subtracting two consecutive timestamps from each other. If the timestamp column is omitted, the time interval can be typed in by the operator. The second is the last read cell (**lastRead**), which is used by the **readStackTonnages** macro, explained next.

Within the **readStackTonnages** macro the following additional variables are generated:

- **timeInterval**. The amount of time that passes between consecutive data points. The value of **timeInterval** is obtained from the parameter with the same name, as indicated at the top of the data sheet.
- **dataPointsToStack**. The number of data points that will need to be read from the data sheet. This is calculated as **stackSimTime / timeInterval**, granted that **timeInterval** isn’t equal to zero.
- **Tonnes (i)**. An array that stores the tonnages fed to the stacker in each time interval. The size of the array is dependent on the amount of data points that need to be read.

---

<sup>5</sup> Slight model adjustments would be necessary to account for varying intervals between data points, as one of the model assumptions was that grade and volume data is available at regular increments



- **Cubes (i).** As the geometric simulations will make use of volumes instead of masses, the mass in **tonnes(i)** is converted to cubic meters by dividing it by the bulk density.
- **Grades (i).** An array to store the grade of material that is fed to the stacker in each time interval.

The procedure starts by setting all values in the three arrays, **tonnes(i)**, **cubes(i)**, and **grades(i)**, to empty. It then moves the cursor to the cell indicated as the last one that was read into the model. If this is the first time the data set is used, the value of **lastRead** will refer to the first cell in the second column that contains a tonnage. A loop is now executed, an amount of times equal to **dataPointsToStack**, using a counter variable. Steps followed in the loop are shown in Figure 19.

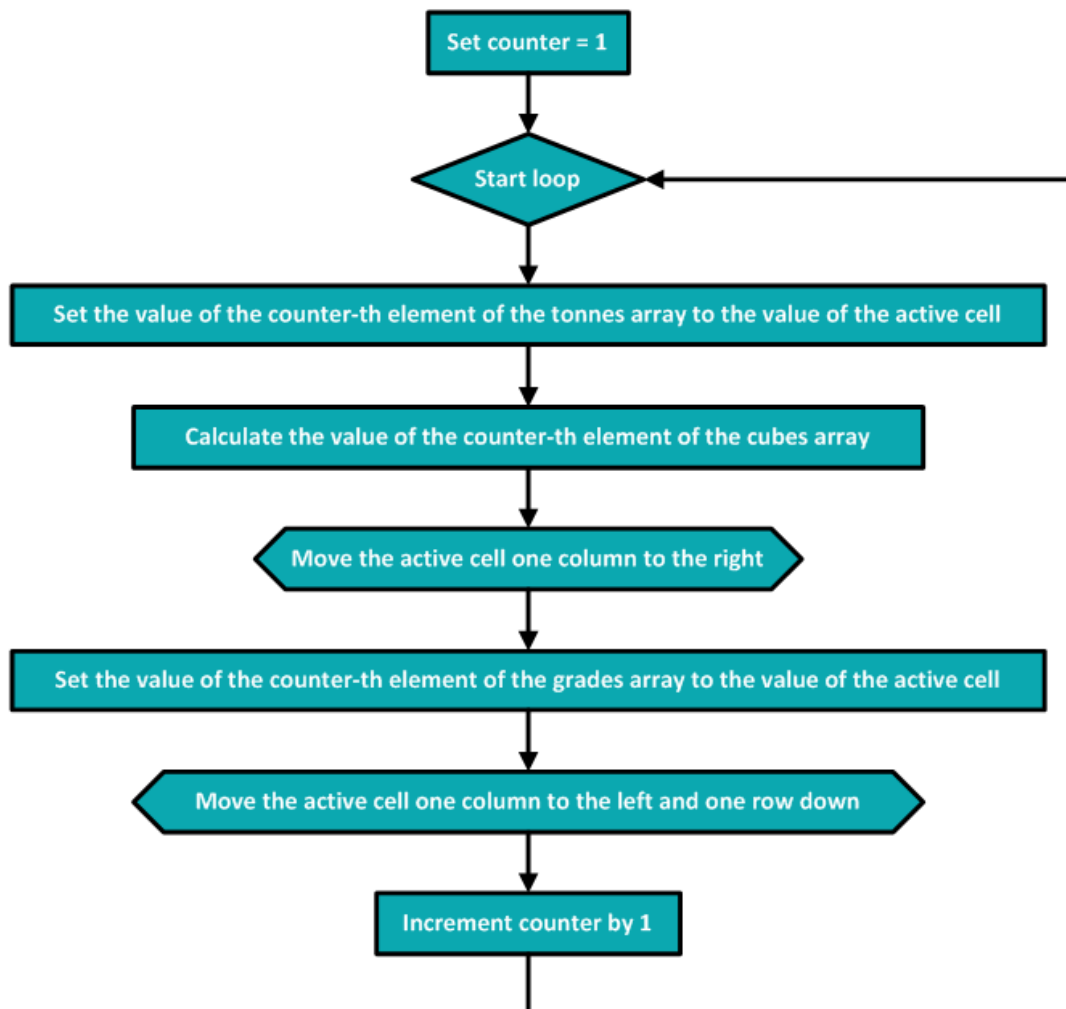


Figure 19: Flowchart of ReadStackTonnages macro

As the procedure moves through the data set during the loop, the cell represented by **lastRead** changes, moving downward row by row. If a simulation finishes and not all data points were used, the value of **lastRead** is written back to the top of the data sheet, so that the next simulation starts reading data points where the previous one finished. The **lastRead** parameter can also be changed manually on the data sheet, should the need arise.

If **stackFromScratch** is set to “Continuous stacking”, the simulation model needs information about the geometry of material that is already on the pile. Therefore, every time a new simulation is started, a macro called **readGeoData** is run. This procedure loops through all the cells on the “**GeoData**” worksheet, and assigns values to the corresponding elements in the **heights** array. The **readGeoData** macro is a reversal of the **fillCells** macro that is discussed in section 4.2.4.

#### 4.2.1. Coneshell method: Stacking the first cone

Two simulation parameters specific to coneshell stacking are introduced at this point:

- The stacker step increment – how far it moves every time a new coneshell is stacked (in degrees): **stackerOffset**
- **StackerOffsetDistance**. This is the actual distance (in meters) that the stacker travels every time it moves a step increment to start stacking a new cone. It can be calculated as the chord of a circle with radius = **rStack** and angle = **stackerOffset**, using Equation 16. The importance of this parameter will be discussed later in the section.

$$\text{Chord length} = 2 \times r \times \sin\left(\frac{\text{angle}}{2}\right)$$

Equation 16: Calculation of a circle's chord length when radius and central angle are known

Now follows one of two core macros in the coneshell stacking part of the model, namely **StackFirstCone**. The procedure stacks the first cone that will form the base of the stockpile, with successive cone-“shells” being stacked next to it as described in section 2.1.1. The **StackFirstCone** macro will, ideally, run only once for every stockpile, and will not execute again as long as **stackFromScratch** is set to “Continuous stacking”. This condition is evaluated in the main **Stack** macro. Four new variables are defined, their values only retained within the scope of the **StackFirstCone** macro. These are:

- **lastRadius** – Radius of the last cone stacked in the previous simulation. This value is stored on the calculation sheet, and read into the model during the **defineEverything** macro.
- **rSmall** – Radius of the last cone stacked in the current simulation.
- **hSmall** – Height of the last cone stacked in the current simulation.
- **volumeSmallCone** – Volume of the last cone stacked in the current simulation.
- **dataPoint** – A global integer that keeps track of which data points have been read from the **cubes** and **grades** arrays. It is initialized with a value of one, to indicate that stacking should start with the first **cubes** value.
- **Dis** – Horizontal distance between any point and the current position of the stacker
- **rNext** – Radius of the cone currently being stacked

The function of the variables above will become clear during the explanations to follow.

*Part I: Stack the first cone, equal in volume to the first data point*

For this part of the code to execute, **lastRadius** has to be equal to zero, which serves as a way for the model to confirm that no previous cones have been stacked. The procedure starts by looking at the first value in the cubes array, and determining the radius needed to stack a cone of equivalent volume.

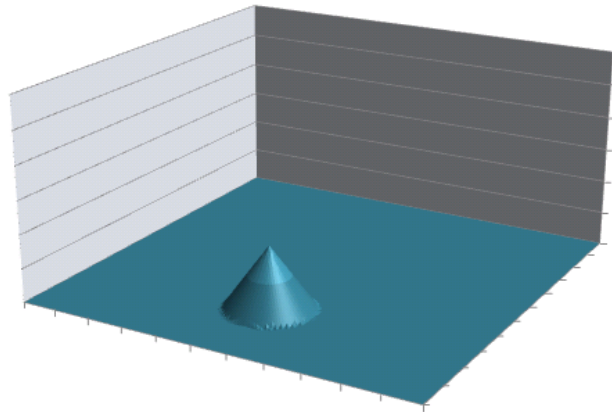


Figure 20: Model output - stacking the first cone

If Equation 11 for the volume of a cone is combined with the trigonometric relationship between cone radius and cone height shown in Equation 15, the following is obtained:

$$V = \frac{1}{3}\pi r^3 \tan(aRepose)$$

Equation 17: Cone volume expressed in terms of radius and repose angle

From Equation 17 one can derive a formula to describe the radius of a cone given the volume:

$$r = \sqrt[3]{\frac{3V}{\pi \tan(aRepose)}}$$

Equation 18: Radius of a cone when volume is known

The required cone radius is calculated by using Equation 18, and the value assigned to **rSmall**. Consequently **hSmall** can be calculated from Equation 15, and **volumeSmallCone** from Equation 11.

The procedure now loops through the coordinates on the base of the stockpile by using a counter, in the same way as described for the **CreatePoints** macro in section 4.1. For each coordinate its distance from the point where the stacker is discharging is calculated. This is done by making use of the polar coordinates of the stacker (**rStack**, **aStack**), the polar coordinates of the current point (**radial(counter)**, **angular(counter)**), and Equation 10. If the calculated distance is smaller than the radius of the cone, it means that the coordinate falls within the stacked cone. The next step is to determine the height of the cone at said coordinate.

The height determination again uses the trigonometric relationship between cone height and cone radius. Once a coordinate's distance from the cone midpoint (i.e. stacker discharge point) is known, it is subtracted from the cone radius to calculate the height of the cone at that coordinate (see Figure 21).

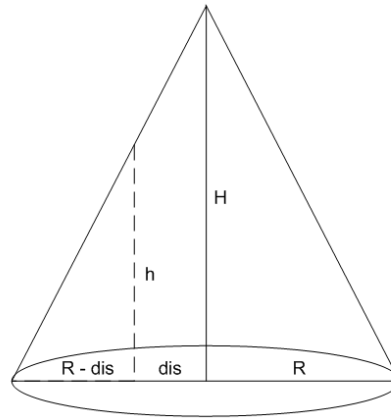


Figure 21: Height of a cone given distance from midpoint

$$heights(counter) = (rSmall - dis) \times \tan(aRepose)$$

Equation 19: Height of a cone given distance from midpoint

The result of Equation 19 is written to the **heights** array for the relevant **counter** value. In this way heights are assigned to every coordinate on the stockpile base, which will be used to draw the pile in a later procedure.

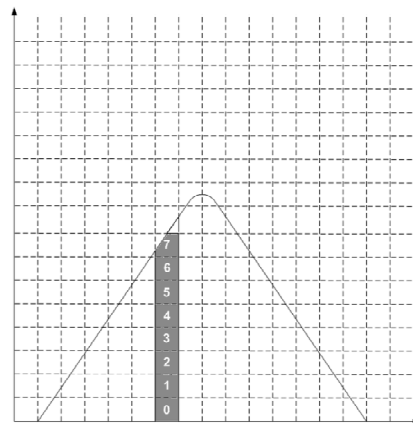


Figure 22: Assigning grade values to the first cone

Before moving on to the next coordinate, the grade value of the first data point needs to be assigned to every coordinate that falls within the cone boundaries. Therefore, after it has been established that a coordinate falls within the cone, and the height has been determined, the following is carried out: The procedure loops through the z-values in **CartArray** from zero to the height of the pile for the given x- and y-coordinates, assigning the value of **grades(counter)** to each. This is demonstrated graphically in Figure 22.

Finally **counter** is incremented by one, and the value of **dataPoint** is set equal to two. The stacker model is now ready to stack the next data point.

### Part II: Stack cones of increasing size in “shells” over the first

This part of the macro executes in a loop, and will keep on executing while the radius of the cone being stacked is smaller than a full cone radius, and **dataPoint** is smaller than **dataPointsToStack**, i.e. the last data point hasn't been reached.

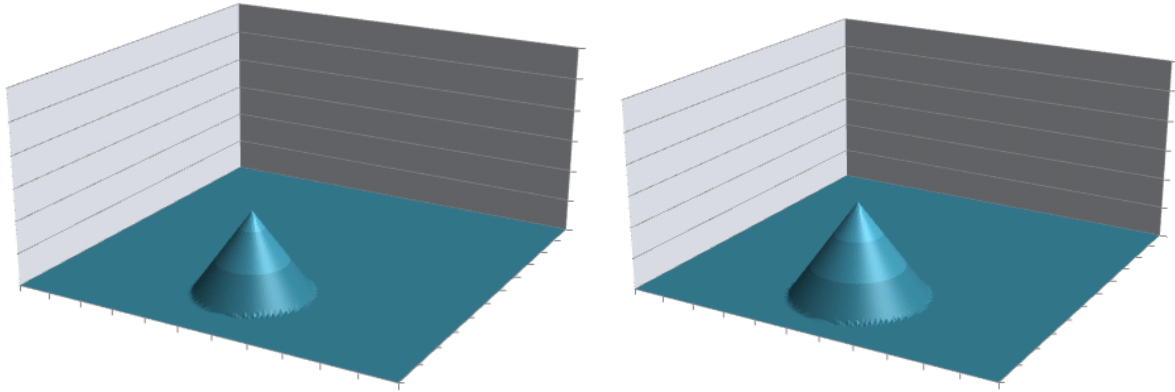


Figure 23: Model output - stacking successive cones

From Part I of this macro **rSmall** is the radius of the very first cone stacked, and **hSmall** and **volumeSmallCone** have been calculated accordingly. In order to form a shell over this small cone, a bigger cone is stacked over it. The volume of the shell is then equal to the volume of the small cone subtracted from the volume of the big cone. This needs to equal the volume contained in the next data point, i.e. the next element in the **cubes** array. In equation form, combined with Equation 17:

$$cubes(dataPoint) = \frac{1}{3}\pi r_{Next}^3 \tan(aRepose) - volumeSmallCone$$

Equation 20: Volume of a shell over a smaller cone

From Equation 20 the radius of the next cone to be stacked is:

$$r_{Next} = \sqrt[3]{\frac{3(cubes(dataPoint) + volumeSmallCone)}{\pi \tan(aRepose)}}$$

Equation 21: Radius of conical shell

Note that, since the radius of each consecutive cone is dependent on the number of cubes fed to the stacker in the specified time interval, the conical shells will all differ in thickness. If the calculated radius is smaller than a full cone radius (**rCone**), a loop is executed as in Part I. Coordinates are evaluated by calculating their distance from the stacker discharge, calculating the height of the cone at different points, and assigning grade values to all blocks within the cone. The only difference in Part II is that before a grade value is assigned to a block, the procedure first checks to make sure that there isn't already a grade value assigned from a previous cone. After a few shells have been stacked the profile of grades represented would resemble Figure 24.

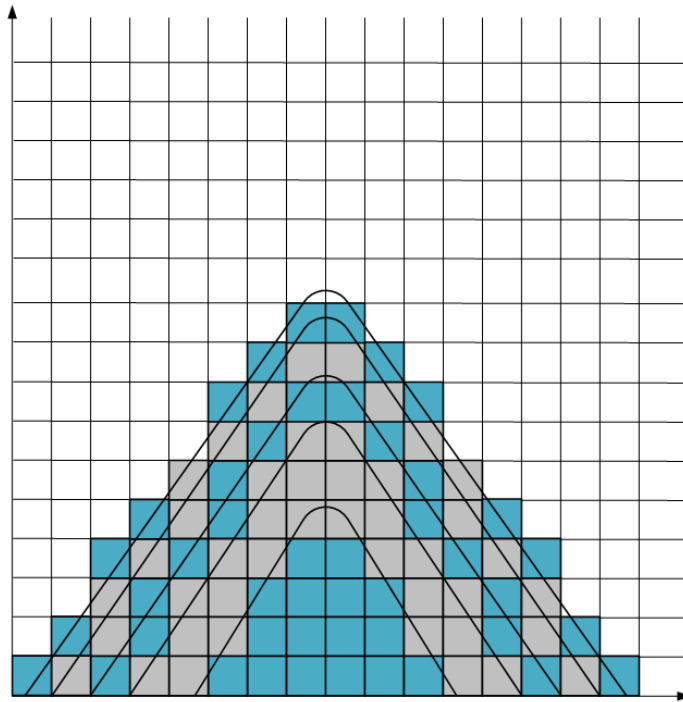


Figure 24: Blocks filled with grades as successive conical shells are stacked

The last steps are to set **rSmall** equal to the current value of **rNext**, and recalculate **hSmall** and **volumeSmallCone**. The value of **dataPoint** is also incremented by one before returning to the start of Part II and stacking the next conical shell.

### *Part III: Stack a final conical shell*

As noted previously, Part II will execute until either the data points run out or **rNext** increases to **rCone**. If the former occurs, the procedure will end without executing Part III. In the latter case though, the following will happen: With every new data point read from **cubes**, the value of **rNext** will increase incrementally. The moment **rNext** increases to be larger than **rCone** though, Part II will end without stacking the shell. This means that the last conical shell that was stacked will be slightly smaller than a full cone. In order for the first stacked cone to be complete a final shell with a radius exactly equal to **rCone** is stacked. This is the function of Part III.

The last conical shell is stacked in exactly the same way as the previous ones, except that no radius is determined. The value of **rNext** is set equal to **rCone**. After the shell is stacked its volume is deducted from the volume contained in the current element of the **cubes** array. This is done because some of the material from the current data point may not be needed in the stacking of the last shell, and should therefore be preserved for the next procedure in the stacker model. For the same reason the value of **dataPoint** is not incremented, to make sure that the next procedure starts with the same element in the **cubes** array.

Since the stacker has, at the end of Part III, stacked a full cone, it needs to move for the first time. This is done by incrementing the value of **aStack** with the value of **stackerOffset**. The procedure makes sure that

after this incrementing the value of **aStack** doesn't exceed 360 degrees and that if it equals 360 degrees the value is reset to zero. This value of **aStack** is written to "CalcSheet" and stored to be used in the next procedure. The value of **rSmall** is also written to "CalcSheet" to be read back into the model at a later stage as **lastRadius**. If Part III executed **rSmall** will be equal to zero, else it will equal the radius of the last cone stacked.

#### 4.2.2. Coneshell stacking: Stacking consecutive shells

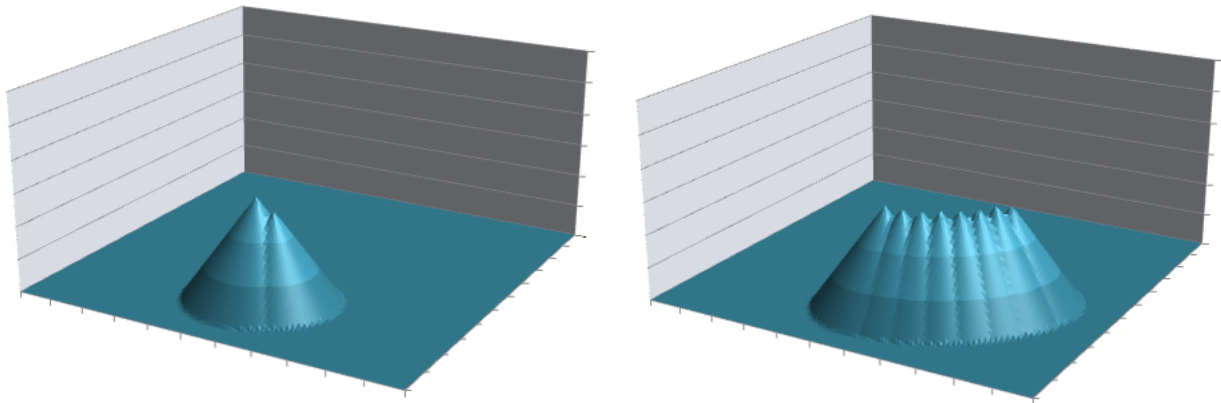


Figure 25: Output of model after stacking consecutive coneshells

The second major macro within the coneshell stacking model is called **StackShells**. Most of the **StackShells** procedure is concerned with determining the volume of a coneshell. As no formulas could be found in literature geometrical approximation was used. The method will be explained at the hand of Figure 26.

#### *Finding the coneshell volume*

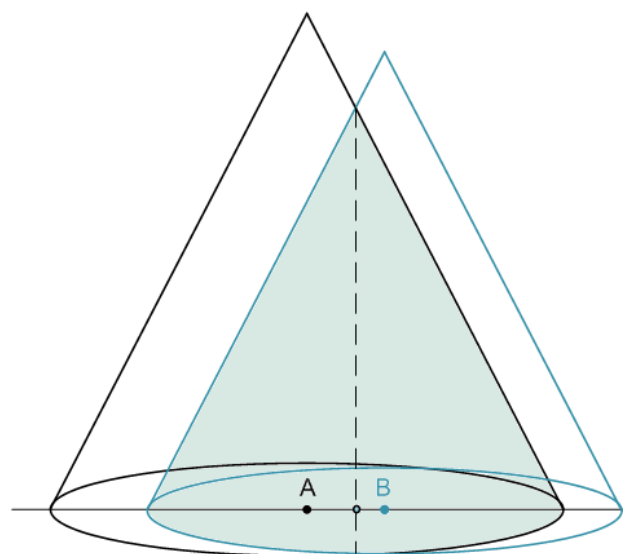


Figure 26: Method for calculating the intersecting volume of two cones

The volume of intersection between two cones can be approximated as the volume of another cone, with midpoint directly under the point of intersection. This point will always lie between the midpoints of the first two cones. If the first, larger cone has midpoint “A”, and the second cone “B”, the distance between the midpoint of the intersection cone (M) and the other two midpoints are “a” and “b” respectively. By inspection the radius of the intersection cone can be found as:

$$r_M = r_B - b = r_A - a$$

Equation 22: Determination of intersection cone radius

Also by inspection, the relationship between a and b is:

$$b = AB - a$$

Equation 23: Relationship between a and b

In the equation above AB is equal to the offset between the two cones. By substituting Equation 23 into Equation 22:

$$a = \frac{(r_A - r_B + AB)}{2}$$

Equation 24: Calculation of a

After finding the value of a, b and  $r_M$  can be calculated by using Equation 23 and Equation 22 respectively. Once the radius of the intersection cone is known its volume can be calculated. The volume of the coneshell is calculated as the volume of the intersection cone subtracted from the volume of cone B.

The method described above is executed in a macro called **ConeShellVolume**. The procedure needs one variable as input; the radius of the second cone. Within the macro variables for **a**, **b**, and the **shellHeight** are defined, as well as the three geometric parameters for the interception cone, namely **interceptRadius**, **interceptHeight**, and **interceptVolume**. It is not necessary to specify the radius of cone A, since during normal operation it will always be equal to the full cone radius, i.e. **rCone**.

The results of Equation 22, Equation 23, and Equation 24 are calculated as follows:

$$a = \frac{(rCone - shellRadius + stackerOffsetDistance)}{2}$$

$$b = stackerOffsetDistance - a$$

$$interceptRadius = rCone - a = shellRadius - b$$

Equation 25: Calculations in ConeShellVolume



The values of **shellHeight** and **interceptHeight** are found using the trigonometric relationship between radius and height, and **interceptVolume** is calculated using Equation 11. Finally the coneshell volume is calculated as:

$$coneShellVolume = \frac{\pi \times shellRadius^2 \times shellHeight}{3} - interceptVolume$$

Equation 26: Calculation of coneshell volume

### Stacking shells

The **ConeShellVolume** macro is used iteratively in **StackShells**, which will be explained next. In order for this procedure to model the way coneshells are formed one must first investigate what happens in the intersection of two cones.

In Figure 27 two cones are intersecting, with their midpoints offset by a distance equal to **stackerOffsetDistance**. If the large cone's radius is equal to **rCone**, the small cone's radius can be calculated as **rCone – stackerOffsetDistance**. With the intersection as shown no coneshell is formed, as the small cone is wholly contained within the other. This would also be the case for any cones smaller than this critical radius. Shells only start forming once the radius of the small cone is increased from (**rCone – stackerOffsetDistance**).

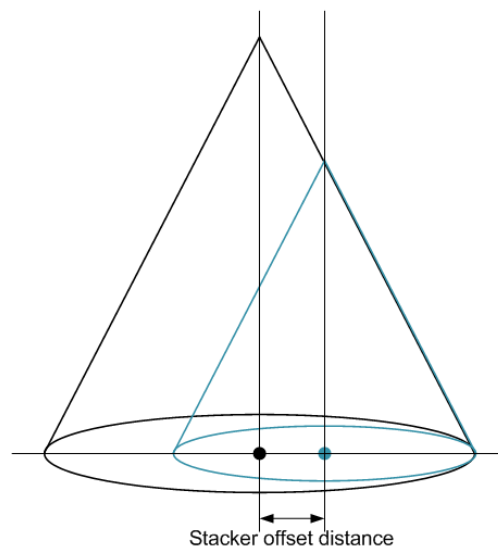


Figure 27: Two cones intersecting

For the reason explained above the first step in the **StackShells** macro is to make sure that the value of **lastRadius** is larger than the critical radius where shells start forming. If this is the very first time the procedure is executed (i.e. a new stockpile is being constructed) **lastRadius** will equal zero. The value of **rSmall** will then be set to the critical radius to start the procedure. Else the value of **rSmall** is set to equal the last radius stacked in the previous execution.

The position where the stacker is to start discharging material is set to the value of **lastStacked**. This is read from “CalcSheet” during **defineEverything**, after being stored in a previous execution. Much like in the **StackFirstCone** macro, the value of **rNext** has to be calculated for every new data point (i.e. the next element in the **cubes** array). This is done by using Microsoft Excel’s GOALSEEK method, executed within “CalcSheet”.

<b>coneShellVolume</b>	<u>1658.22</u>	<b>shellRadius</b>	<b>19.10</b>
<b>stackerOffsetDistance</b>	4.88	<i>shellHeight</i>	14.92
<b>rCone</b>	19.84	<i>interceptRadius</i>	17.0
<b>aRepose</b>	0.663	<i>interceptHeight</i>	13.3
<i>a</i>	2.81	<i>interceptVolume</i>	4039
<i>b</i>	2.07		

Figure 28: GOALSEEK method executed to find shell radius

Figure 28 shows how GOALSEEK calculates the required value of **rNext**. The parameters printed in **bold** have been written to “CalcSheet” from within the stacker model. The values of all other parameters are calculated as previously explained.

Coneshells are built in the same way as the conical shells were built in **StackFirstCone**. The radius is increased by small increments so that each shell is equal in volume to the value that is read from the **cubes** array. When a coneshell reaches its full size, i.e. **rNext** = **rCone**, the stacker moves to its next position and the procedure is repeated. Therefore the calculated value of **coneShellVolume** is equal to the total volume of the current shell being stacked, which includes all the smaller shells previously stacked in the same position. Put differently, **coneShellVolume** is equal to the volume of the previously stacked coneshell plus the current **cubes** value.

$$coneShellVolume(rNext) = coneShellVolume(rSmall) + cubes(dataPoint)$$

Equation 27: Target value for coneShellVolume

When the GOALSEEK procedure shown in Figure 28 executes, the value of **shellRadius** is adjusted until **coneShellVolume** reaches the target value shown in Equation 27. The solution value of **shellRadius** is read into the model as **rNext**. The value of **rNext** is evaluated to make sure that it is smaller than the radius of a full coneshell. The steps that follow are similar to the steps used to stack material in **StackFirstCone**. Coordinates are evaluated based on their distance from the stacker discharge point, and subsequently heights are assigned to each coordinate. Some coordinates will already have height values assigned at this

stage. The height calculated by the **StackShells** macro will only be assigned to a coordinate if the current height value is less than the calculated value. This is synonymous to stacking a layer of material on top of another, without taking any of the previous material away. Grades are also assigned to blocks in the same way as explained previously (see Figure 24). Finally the value of **dataPoint** is incremented, and **rSmall** assumes the value of **rNext**. The model is now ready to stack the next shell. Note that for this scenario the next shell will be stacked in the same position, i.e. the stacker position is not incremented.

If the evaluation finds that **rStack** is larger than a full coneshell radius (**rCone**), a full shell is stacked and the volume of leftover material stored in the relevant element of the **cubes** array for the next execution. The stacker is also moved forward by an angle equal to **stackerOffset**, and **rNext** and **rSmall** are reset to the minimum value (see explanation accompanying Figure 27). The model is now ready to stack a new shell in the next stacker position. The **StackShells** macro will continue executing until all the volumes contained in the **cubes** array are used. It then ends with writing the last stacker position and last shell radius to “CalcSheet”, from where it will be read for the next simulation.

#### 4.2.3. Chevcon stacking: Stacking layers

The chevcon stacking model also makes use of the **StackFirstCone** macro to start a new stockpile, and then adds chevcon layers onto it with the **StackLayers** macro, introduced below. Firstly, some simulation parameters need to be assigned values on the “GUI” worksheet for chevcon stacking:

- The angle through which the stacker boom rotates to stack a full chevcon layer – **aRotation**
- The stacker speed in m/min. This is used to calculate the time per stacker pass (**timePerStackerPass**) in hours as follows:

$$\text{Length of stacker pass} = 2\pi r\text{Stack} \left( \frac{a\text{Rotation}}{360^\circ} \right)$$

Equation 28: Length of a stacker pass in chevcon mode

$$\text{TimePerStackerPass} = \frac{\text{Length of a stacker pass}}{60 \times \text{stacker speed}}$$

Equation 29: Calculation of time per stacker pass

- The number of layers exposed to the reclaimer face. This number is used to calculate the angle through which the stacker needs to rotate every time it is adjusted to start stacking a new set of chevcon layers. The calculation of **moveIncrement** is shown in Equation 30 below.

•

$$\text{moveIncrement} = \frac{\text{Length of a stacker pass}}{(\text{number of layers})(\pi r\text{Stack})} \times 360^\circ$$

Equation 30: Movement increment angle of the stacker boom during chevcon mode (Holderbank, n.d.)

- The increment used for iterative calculations of layer thickness (**increment**). The smaller this increment, the more accurate calculations will be. Unfortunately cutting the calculation increment in half also means that the simulation processing time will be doubled.

Other variable values calculated during **defineEverything**:

- **incrementDistance**. Calculated from **increment** and **rStack**, using Equation 16.
- **incrementHeight**. The vertical distance that the stockpile crest moves downwards when the stacker rotates by an angle equal to **increment**. This is calculated by using ratios. Since the stacker moves vertically from stacking the maximum height of the pile to ground level during a full chevcon layer, the height difference from moving an increment angle is:

$$\text{incrementHeight} = h_{\text{Stockpile}} \left( \frac{\text{increment}}{a_{\text{Rotation}}} \right)$$

Equation 31: Calculation of increment height

- **incrementsInPass**. The number of calculation increments in a stacker pass, i.e. **aRotation** divided by **increments**.
- **moveHeight**. Calculated in the same manner as **incrementHeight**, but using the **moveIncrement** angle.

### *The StackLayers macro*

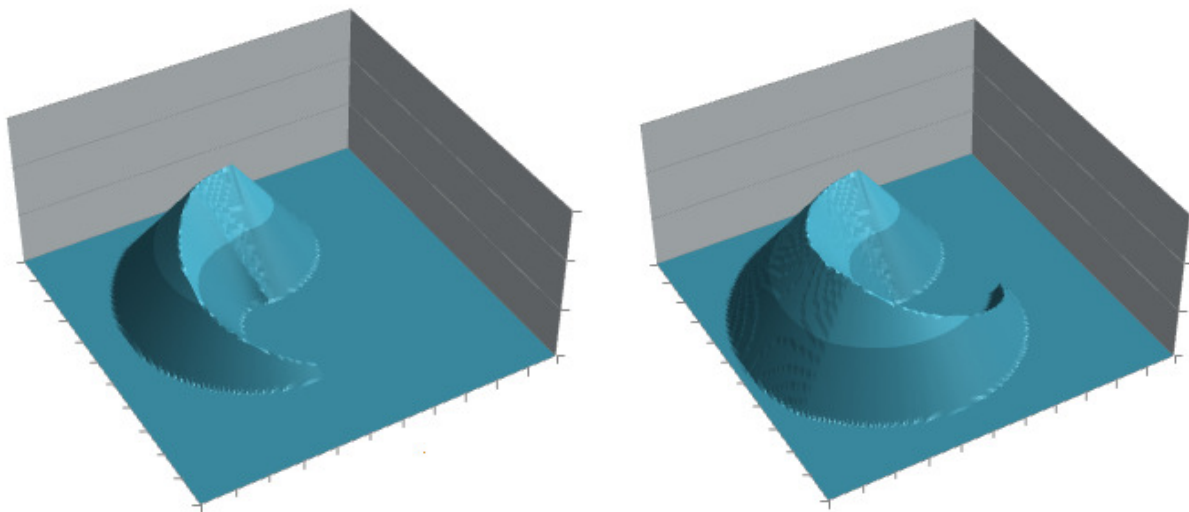


Figure 29: Output of model after stacking chevcon layers

The procedure starts by defining a variable named **maxHeight**, and reading its value from “CalcSheet”. This variable is synonymous with **lastRadius** in coneshell stacking, and is used to pick up where the previous simulation left off. The value represents the maximum height of the last chevcon layer to be stacked. The value of **aStack**, the starting position of the stacker boom, is also read from “CalcSheet”. The value of **aRotation** is added to **aStack** to get **endAngle**, the furthest position that the stacker travels during a pass before reversing.

The next steps in the chevcon stacker model are best demonstrated using a flowchart, as they consist of several concentric loops. A flowchart of the StackLayers macro is shown in Figure 30. Each step shown will be explained below.

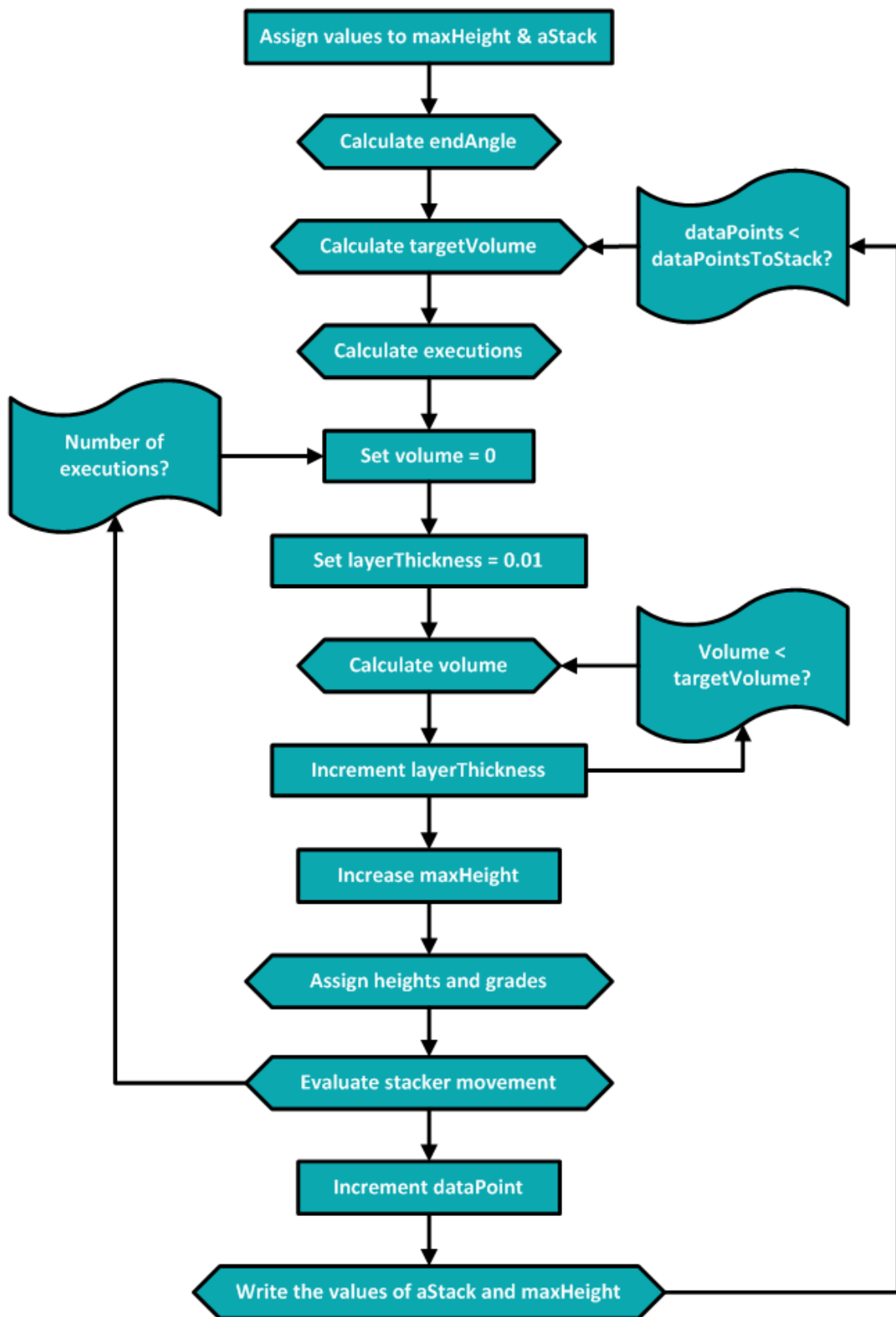


Figure 30: Flowchart of StackLayers macro

**Calculate target volume.**

This is the volume of the next layer to be stacked. The volume is dependent on the layer thickness, which in turn is dependent on the material discharged during each stacker pass. Since the stacker moves at constant speed, even when there is no material on the stacker boom conveyor, the number of layers stacked in any given time period will be constant, with variation in thickness only. The variable **targetVolume** is calculated as the element of **cubes** for the current **dataPoint** divided by the **timeInterval** between data points, multiplied by the **timePerStackerPass**.

$$targetVolume = \frac{cubes(dataPoint)}{timeInterval} \times timePerStackerPass$$

Equation 32: Calculation of target volume

**Calculate executions**

The number of times this part of the code needs to execute is equivalent to the number of layers stacked out of the next **cubes** element. It is therefore calculated as **cubes(dataPoint)** divided by **targetVolume**.

**Set volume = 0**

The variable named **volume** is used to store the total volume of the current layer. This is increased until it reaches the **targetVolume**. The execution starts with **volume** being set equal to zero.

**layerThickness = 0.01**

The current thickness of the layer being stacked. **LayerThickness** is calculated iteratively, incremented by 0.01m, until the calculated **volume** reaches the **targetVolume**.

**Calculate volume**

The volume of a layer can be estimated as the shell formed between two variable-height triangular prisms (see Figure 31). This shell can be cut up into slices, forming triangular shapes as shown. In each shape the larger triangle has dimensions of base and height as **bLarge** and **hLarge** respectively. The smaller triangle is **hSmall** high with a base of **bSmall**. If the shape has a thickness of **incrementDistance**, its volume can be expressed as:

$$Volume = \frac{1}{2} incrementDistance (bLarge \cdot hLarge - bSmall \cdot hSmall)$$

Equation 33: Calculation of increment volume

The volume of a shell will be the sum of the volumes of all the increment shapes, i.e.:

$$\sum_{i=1}^{incrementsInPass} \frac{1}{2} incrementDistance(bLarge_i, hLarge_i - bSmall_i, hSmall_i)$$

Equation 34: Volume of a chevcon layer

In Equation 34  $hSmall_i$  is equal to **maxHeight**, with the value being decreased by **incrementHeight** with every increment. Therefore  $hSmall_{incrementsInPass}$  will be equal to zero. Once **hSmall** is known the rest of the variables in Equation 34 can be calculated as follows:

$$bSmall = \frac{2hSmall}{\tan(aRepose)}$$

$$bLarge = bSmall + 2layerThickness$$

$$hLarge = \frac{1}{2} bLarge \times \tan(aRepose)$$

Equation 35: Calculation of volume parameters

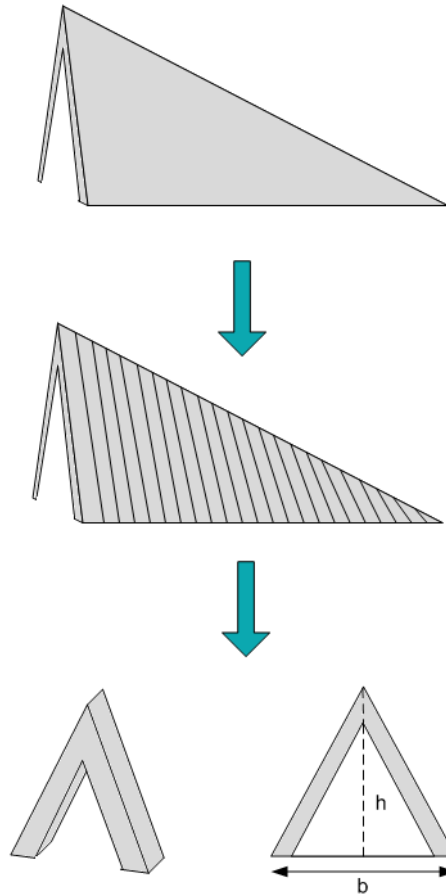


Figure 31: Formation of chevcon volume increments

**Increment layerThickness, Increase maxHeight**

The value of layerThickness is incremented by 0.01, the volume recalculated, and the value of maxHeight is increased to the maximum height of the last layer stacked.

**Assign heights and grades**

The heights of the stockpile for given coordinates are calculated, and grades are assigned as discussed in previous stacking procedures. The width of the base of the stockpile at any point is defined as **base**. The value of **base** varies from its maximum value when the stockpile is at **maxHeight**, to almost zero. The far tip of the chevcon layer isn't perfectly sharp though, so it was assumed that the base at the tip is about 10% as wide as the base at the start of the layer. The calculation of **base** then becomes a linear interpolation between these two points:

$$base = \frac{2maxHeight}{\tan(aRepose)} \left( \frac{0.9(endAngle - angular(counter))}{aRotation} + 0.1 \right)$$

Equation 36: Calculation of base width for a chevcon layer

Once the base width of the triangular cross-section of any part of the pile is known, the height of a point on that cross-section can be calculated by using trigonometry, as shown in Figure 32, by using Equation 37. The **heights** array can therefore be filled by values by looking at firstly the **angular** component in Equation 36 and then the **radial** component in Equation 37 for each counter value.

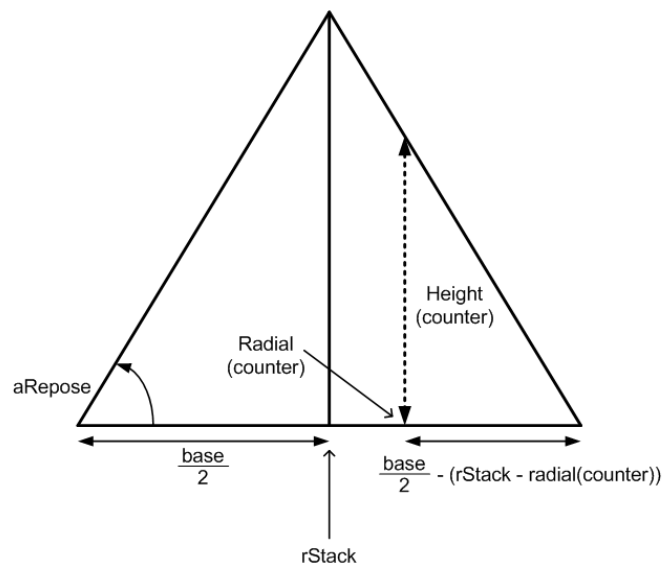


Figure 32: Calculation of height of a chevcon layer

$$height = \frac{1}{2} base \times \tan(aRepose)$$

$$height(counter) = \tan(aRepose) \left( \frac{1}{2} base - |rStack - radial(counter)| \right)$$

Equation 37: Calculation of height of a chevcon layer



***Evaluate stacker movement***

The value of **maxHeight** is evaluated against **hStockpile** to see whether the layer has reached the full stockpile height. If so, the stacker needs to move to the next position. The value of **aStack** is increased by **moveIncrement**, **endAngle** is recalculated, and **maxHeight** is decreased by the value of **moveHeight**.

Finally the value of **cubes(dataPoint)** is decreased by the volume of the last stacked layer, and the next execution can start.

***Increment dataPoint***

The value of **dataPoint** is incremented before starting the outer loop again at the calculation of **targetVolume**.

***Write the values of aStack and maxHeight***

The last stacker position and layer height are written to “CalcSheet” to be read into the model during the next simulation.

**4.2.4. Drawing the stockpile**

The last macro in the stockpile model is called **FillCells**. The purpose of the macro is to write the values contained in the **heights** array to the “GeoData” sheet, from where a three-dimensional plot can be drawn.

Firstly the x- and y-axis is drawn by writing values from (-stockpile radius) to (+stockpile radius) in the first row and first column respectively. Thereafter the procedure loops through the cells as shown in Figure 18 while simultaneously reading from the **heights** array and writing values to the relevant cells.

On the “GUI” sheet one of Microsoft Excel’s preset three-dimensional graphs is drawn from the data in “GeoData”. With every execution of the stacker or reclaimer model the values in “GeoData” are updated and automatically reflected on the three-dimensional plot.

### 4.3.Reclaiming at an angle

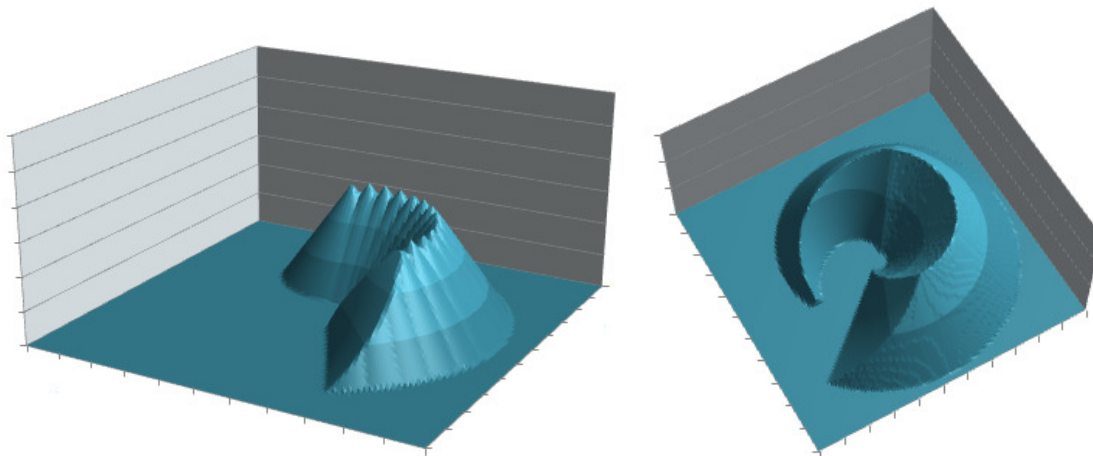


Figure 33: Output of reclaimer model

The reclaimer model remains the same regardless of which stacker model is used. It also introduces a new sheet to each workbook by the name of “Reclaimed”. This sheet is used to store information about material that is reclaimed from the pile, including the number of cubic meters and average grade reclaimed for every time interval.

The reclaimer model uses some of the same macros already introduced in the stacker model sections, like **defineEverything**. Additional variables that are read from the “GUI” sheet are:

- **rakeAngle**. The angle between the reclaimer harrow and the horizontal. This is usually designed to equal the angle of repose of the material.
- **reclaimIncrement**. The value of the reclaimer calculation increment (in degrees). As noted before, a decrease in the calculation increment results in higher accuracy, but this is offset with higher processing time.
- **reclaimSimTime**. The time (in hours) that the reclaimer should simulate material reclamation before stopping.

Additional parameters defined for use in the reclaimer model:

- **reclaimIncrementDistance**. Similar to **stackerOffsetDistance**. Calculated from Equation 16 by using **reclaimIncrement** and **rStack**.
- **aReclaim**. The last position of the reclaimer, read from “CalcSheet” in order to pick up where the previous simulation left off.

The reclaimer model also makes use of the **ReadGeoData** macro to ascertain the geometric shape of the stockpile, and stores this information in the **heights** array. This step is followed by an almost identical copy of the **readStackTonnages** macro named **readReclaimTonnages**.

The time interval used when reading reclaimer data into the model will depend on the method of data generation for the relevant application. Although it can be the same as the interval used for stacker data, a separate value is used in order to accommodate a difference. The value of **dataPointsToReclaim** is determined by dividing **reclaimSimTime** by **timeInterval**. Two additional 1000-element arrays, **reclaimTonnes** and **reclaimCubes**, are also defined. As was the case when reading data into the stacker model, a variable is created to keep track of where the last point of data entry was. This variable is stored as a cell index called **reclaimLastRead**. The procedure concludes by executing a loop much like that shown in Figure 19, reading data into **reclaimTonnes** and calculating **reclaimCubes**.

The core macro in the reclaimer model is called **ReclaimSlices**. A reclaimer slice can be approximated as a triangular prism with height dependent on the rake angle and stockpile height, base as wide the stockpile, and thickness equal to the calculation increment (**reclaimIncrement**).

Additional variables are defined within the scope of the **ReclaimSlices** macro as follows:

- **startAngle**. The angular coordinate of the first point at which the specified slice starts cutting into the stockpile
- **reclaimLengthAngle**. The difference in angular coordinates between the closest and furthest points at which the reclaimer starts cutting into the stockpile, i.e. points A and B in Figure 34. The value of this variable is dependent on the rake angle and stockpile dimensions, calculated as follows:

$$reclaimLengthAngle = 2\arcsin\left(\frac{hStockpile}{2rStack \times \tan(rakeAngle)}\right)$$

Equation 38: Calculation of reclaimer length angle

- **r**. A counter variable used to keep track of the number of reclaimer slices.
- **slice**. The z-coordinate at which the reclaimer cuts the stockpile for any combination of x- and y-coordinates.
- **highestPoint()**. An array with 1000 elements. It is used to, for every increment that the reclaimer moves, calculate the highest point at which the reclaimer cuts the pile. The height is needed to calculate the volume of a slice, and would usually equal the maximum stockpile height. It is created as a variable however, to account for non-standard operation where some parts of the stockpile might not be stacked to the full height.
- **tempGrade()**. An array used to store the individual grades that make up the blend of each reclaimer slice. The size of the array is set arbitrarily large to accommodate multiple grade values.
- **GradeBlocks**. The number of grade values stored for each reclaimed slice

Arrays are defined to store the number of cubic meters reclaimed and the average grade of each reclaimer slice. These are called **reclaimSliceCubes** and **reclaimSliceGrade**, both with number of elements equal to **dataPointsToReclaim**. The procedure to follow is executed in a loop for every data point that is reclaimed.

By using the **counter** variable to move through the points in the three-dimensional matrix in the same way as was done in the stacker models, each coordinate is evaluated to determine if it is reclaimed. Firstly the procedure looks at the value of **angular(counter)**. If this value falls between **startAngle** and (**startAngle** + **reclaimerLengthAngle**), i.e. in the path of the reclaimer, the procedure moves to the next step, where the value of **slice** for the particular coordinate pair is calculated.

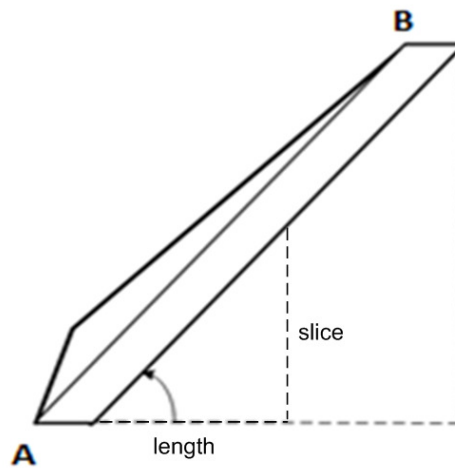


Figure 34: Calculation of slice height

**Slice** can be calculated by using a trigonometric relationship between length (as shown in Figure 34) and **rakeAngle**. In order to calculate length the formula for a chord of a circle is used, with radius equal to **rStack** and angle equal to (**angular(counter) – startAngle**). The combined equation yields:

$$slice = 2rStack \times \sin \left( \frac{angular(counter) - startAngle}{2} \right) \times \tan (rakeAngle)$$

Equation 39: Calculation of slice height

If **slice** is smaller than **heights(counter)** all points above **slice** will be reclaimed. Therefore for every point between  $z = \text{slice}$  and  $z = \text{heights(counter)}$  the grade value needs to be added to the blend that will determine the average grade for the particular slice. Each reclaimed grade is added to the **tempGrade** array, and counted by incrementing the value of **gradeBlocks**. The value of **heights(counter)** is also evaluated to determine whether it is the highest point encountered for the current slice. If this is the case, the value of **highestPoint(dataPoint)** is changed to equal **heights(counter)**.

After an incremental slice is reclaimed its value is added to **reclaimSliceCubes(dataPoint)**. The volume of a reclaimed slice is approximated as a triangular prism. This approximation will be relatively accurate as long as the reclaimer calculation increment, i.e. the thickness of the slice, is sufficiently small. The slice volume is then calculated as:

$$\text{slice volume} = \frac{\text{highestPoint}(\text{dataPoint}) \times r\text{Cone} \times \text{reclaimOffsetDistance}}{\sin(\text{rakeAngle})}$$

Equation 40: Calculation of a slice volume

Finally the value of **startAngle** is incremented, following the reclaimer's progression into the pile. If the value of **reclaimSliceCubes(dataPoint)** is smaller than **reclaimCubes(dataPoint)**, the procedure is repeated, reclaiming incremental slices until the volume contained in **reclaimCubes(dataPoint)** is reached. Once the full volume is reached the average grade of the entire reclaimed slice can be calculated. This is done by adding all the grades stored in **tempGrade** and dividing it by **gradeBlocks** to find an average. The value is stored in **reclaimSliceGrade(dataPoint)**, and the values of **tempGrade** and **gradeBlocks** are reset to zero. The **ReclaimSlices** macro can now return to the start of the procedure to stack the next data point, until all data points in **dataPointsToReclaim** have been looped through.

The cubes and average grades for each data point are written to the "Reclaimed" worksheet with the **writeGrades** macro. The data is obtained from the **reclaimSliceCubes** and **reclaimSliceGrade** arrays. By using a **lastWritten** parameter, the address of a cell stored at the top of the "Reclaimed" worksheet, the macro makes sure not to overwrite previous data.

#### 4.4. Forecasting reclaimed grades

Since the primary function of the stockpile simulation model is to aid the user in making decisions about stockpile management, it must be able to provide forecast information. By equipping the user with information about expected grades to be reclaimed from the pile production can be planned accordingly.

A forecast of reclaimed grades is generated by simulating the reclamation of the continuous circular stockpile to calculate grade data for every sector on the pile. For this purpose a macro that executes in the same way as **ReclaimSlices** was written. The **GenerateForecast** macro simulates reclaiming all material that lies in front of the reclaimer, up until the current stacker position. Therefore the loop that controls the reclamation of incremental slices uses the offset between the current stacker and reclaimer positions to guide whether it needs to reclaim another slice. Two additional arrays are created; **gradePerSector** stores grade values that are forecast for each sector, and **countPerSector** keeps track of the number of grade values stored for each sector. Once a slice has been reclaimed the procedure evaluates in which sector of the stockpile the slice is located, and its average grade is added to the relevant **gradePerSector** element. The value of **countPerSector** is also incremented. When all slices have been reclaimed the average grade for each sector can be calculated by dividing **gradePerSector** by **countPerSector**. The forecast grades are

written to a separate part of the “Reclaimed” worksheet by using the **writeForecast** macro, synonymous to **writeGrades**.

## 4.5. User interface

The preliminary user interface worksheet, “GUI”, was replaced with an improved user interface of the same name. Each GUI will be slightly different to cater for application-specific features, but the one presented below has been developed with Colliery X in mind. User interface development is an iterative process, as the user’s needs have to be reassessed regularly in order to ensure full functionality. As was noted by Arnott (2005), the notion that a DSS evolves through an iterative process of system design and use is central to the theory of DSS. The different elements of the user interface will be presented and discussed in this section.

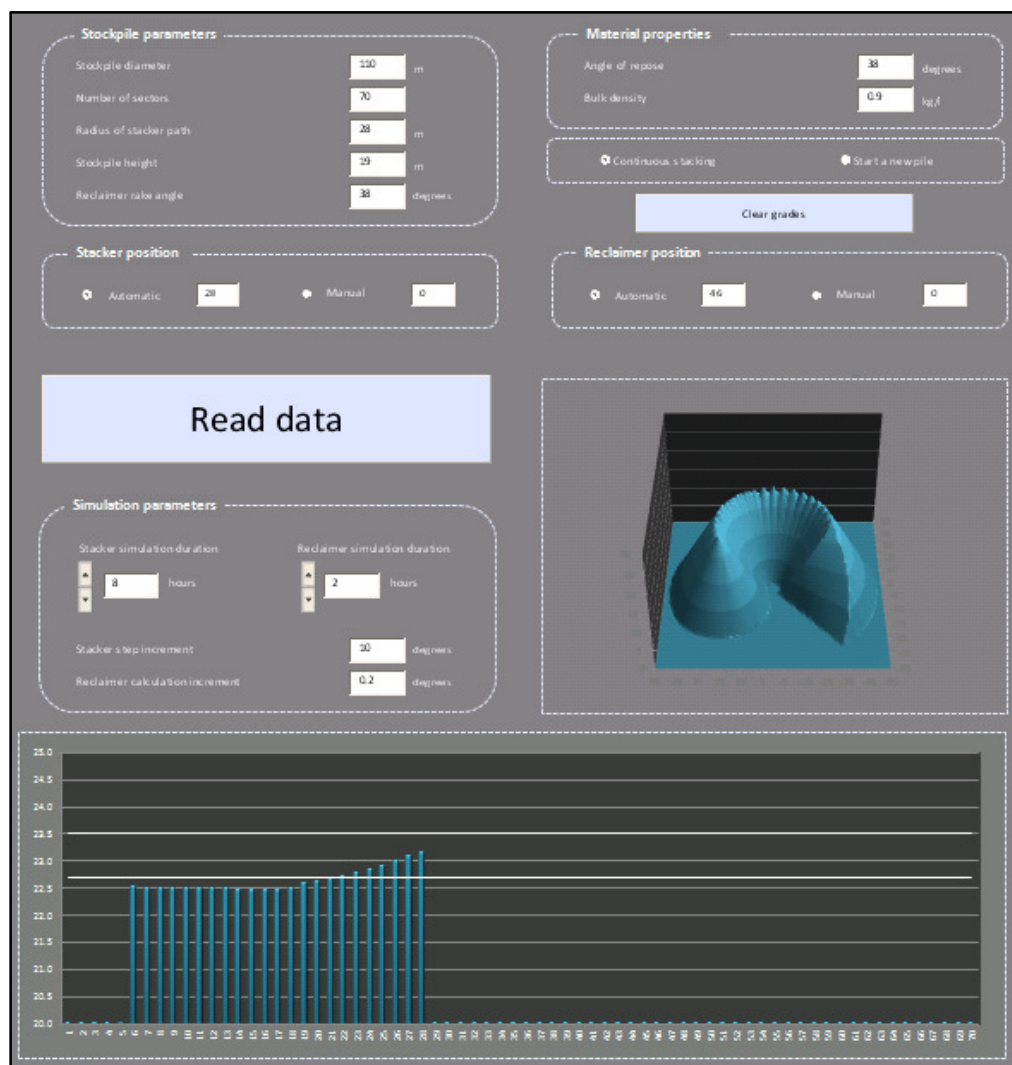


Figure 35: Coneshell stacking full graphic user interface

### 4.5.1. Coneshell stacking

Figure 35 shows the user interface developed for the coneshell stacking model. It can be divided into three parts, each of which will be discussed separately.

Figure 36 is a screenshot of the first part of the coneshell stacking GUI. It provides input space for stockpile parameters and material properties. These parameters were placed at the top of the GUI sheet as they will not be changed often, and should thus not form part of the core simulation section. Also in the top part of the sheet is the radio button controlling whether a new pile is stacked or the current one is continued. During normal operation the option will always be set to “continuous stacking”. A button is also provided to clear grade values from the system when a new pile is stacked. When this button is clicked the user is first prompted with the message “Are you sure you want to clear all grades?” to prevent accidental data deletion. The final elements of the first GUI section are the stacker and reclaimer position controls. If the radio button is set to “Automatic” the position as tracked by the stacker/reclaimer model is used. A “Manual” option is however provided for when the user wants to indicate a different position for either of the machines. A possible future development would see real-time tracking of stacker and reclaimer positions (refer to explanation in section 2.1.4), which would negate the need for these options.

The screenshot shows the Coneshell stacking GUI Part 1. It features four main input sections: **Stockpile parameters** (diameter: 110 m, sectors: 70, radius: 28 m, height: 19 m, rake angle: 38 degrees), **Material properties** (angle of repose: 38 degrees, bulk density: 0.9 kg/l), **Stacker position** (Automatic selected, position: 28; Manual selected, position: 0), and **Reclaimer position** (Automatic selected, position: 46; Manual selected, position: 0). A **Clear grades** button is located between the material properties and position sections. A **Continuous stacking** radio button is selected, and a **Start a new pile** radio button is also present.

Figure 36: Coneshell stacking GUI Part 1

The second part of the coneshell stacking GUI, shown in Figure 37, is concerned with data entry and simulation. The intent is that the data entry process will happen on a daily basis, generating a daily stockpile report which production managers can use in decision-making. Therefore the simulation duration would usually be set to 24 hours on both the stacker and reclaimer sections. Although the reclaimer calculation increment and the stacker step increment are seen as simulation parameters, they will also rarely be adjusted. This means that once production data is input onto the data sheet the user needs only to press the “Read data” button for all macros to be executed. The stacker model, reclaimer model, and forecast section of the simulation run in turn, generating all outputs displayed on the screen. These outputs include a visual representation of the stockpile profile as shown.



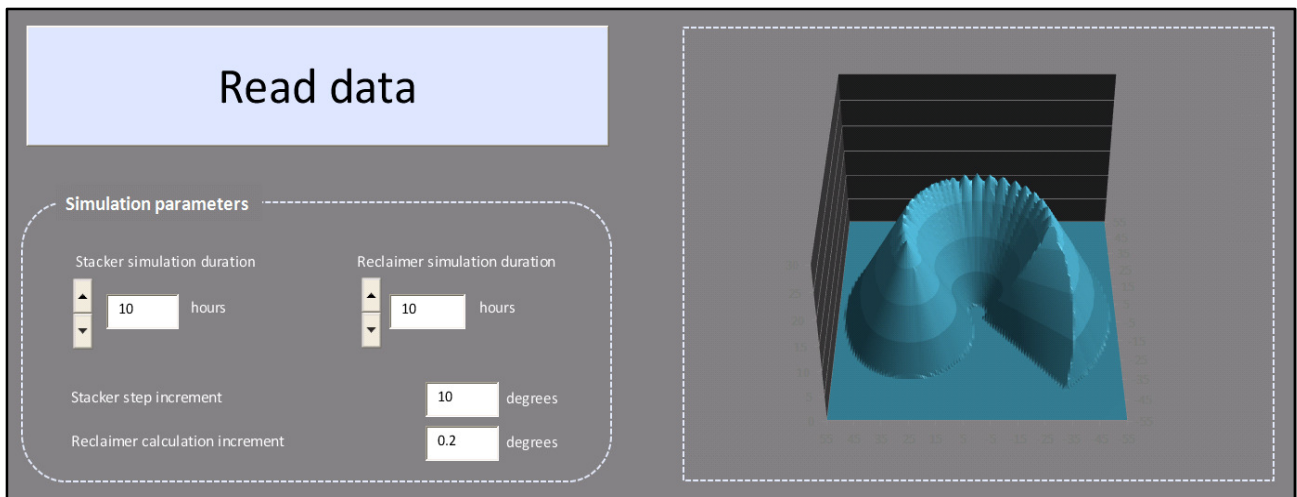


Figure 37: Coneshell stacking GUI Part 2

The last part of the user interface sheet, shown in Figure 38, displays a forecasted grade distribution for the full stockpile volume. It plots average grade versus sector number, and serves to indicate in advance where sub-specification grades will be found. Grade values will of course only be available for the stockpile sections where material is currently stacked.

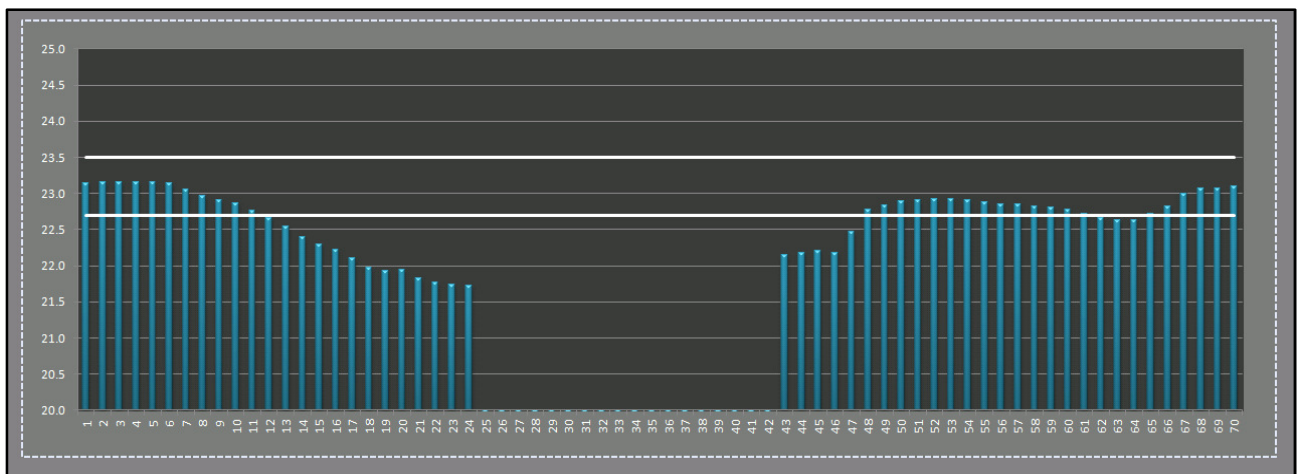


Figure 38: Coneshell stacking GUI Part 3



### 4.5.2. Chevcon stacking

The chevcon stacking user interface is very similar to the one developed for coneshell stacking. The only differences are observed in part two of the sheet, and will be discussed below.

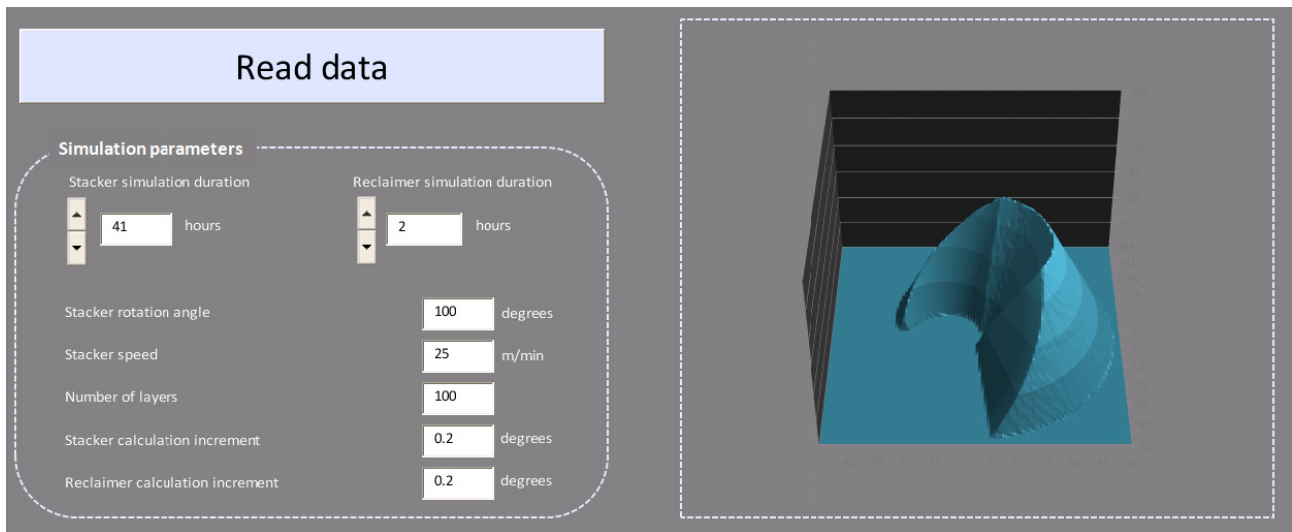


Figure 39: Chevcon stacking GUI Part 2

Part two of the chevcon stacking GUI (Figure 39) makes provision for the entry of additional simulation parameters. These are the stacker rotation angle, stacker speed, number of layers stacked, and the stacker and reclaimer calculation increments. As in the coneshell model the simulation parameters will rarely be changed, so the only action required from the user would be to click the “Read data” button when the stockpile model is to be updated.

## 5. Model validation

This chapter seeks to confirm the validity of the simulations models by comparing predicted output values to known data. Data was taken from a case study conducted at a South African coal processing facility, Colliery X. The data set included information on tonnages per hour, grade results from lab samples, and a manual tracking sheet for stacker and reclaimer positions. Only the coneshell stacking model could be used in validation, since, as mentioned before, Colliery X currently only makes use of coneshell stacking on the stockpile in question.

Attempts were made to obtain chevcon data from other facilities, and two difficulties were encountered. Firstly, the mineral processing industry is highly competitive, and companies are therefore apprehensive about sharing data as specific as hourly product tonnages and grades. Secondly, the type of dataset needed for validation purposes is not commonly available. Most product stockpiles are sampled as material is placed onto the pile. This will of course be sufficient information when the output of the pile is modelled. However, for the purpose of validation, sampling of the output is also needed. In most cases the expense involved in sampling an additional stream is not justified, as the application does not require it. Colliery X presents a unique situation where the stockpile output is sent directly to a customer, making the data needed for validation available.

The accuracy of the model predictions is dependent on a number of factors. Computer scientists often use the term “Garbage-in-Garbage-out” (GIGO) to emphasize the importance of data integrity. This principle applies to any data processing system, of which a simulation model is a good example. The model accuracy will therefore be severely limited by the accuracy of measurement instrumentation, sampling results, and operator inputs. Material properties, like bulk density and angle of repose, can also vary significantly from one day to the next. The values used in the simulation model development are averages and approximations, so it is important to measure and adjust these values on a regular basis.

During normal use of the decision support tool data will be input on a daily basis. Therefore it will rarely need to deal with more than 24 hours worth of data points at a time. For the purpose of this validation though, a period of one month was simulated, in order to observe what happens when the stockpile is stacked and reclaimed continuously. The data set used was chosen at random, bearing in mind that all relevant information (tonnes, grades, stacker/reclaimer positions) for the chosen period had to be available. In the interest of comparability, all simulations carried out in Chapter 5 and Chapter 6 used the same set of data, i.e. production data from December 2011. After measures were implemented to account for difficulties in the validation of the coneshell stacking model, an additional month (August 2012) was modelled. This was done in order to confirm that the new measures were effective in their goal of increasing the accuracy of validation.

## 5.1. Coneshell model

### 5.1.1. Accuracy of geometric modelling

The first step in confirming the validity of the coneshell stacking model was to investigate whether it could predict how the stacker and reclaimer move around the pile. This would give an indication of how accurate the geometric approximations used have proven to be.

Figure 40 shows the result when the stacker position as predicted by the simulation model is plotted versus the position noted by operators on the quality control sheet. The movement of the stacker through sectors is plotted against the cubic meters of material stacked. As can be expected for the modelled graph, a linear relationship exists between the distance the stacker has moved and the amount of material that has been stacked. For most of the graph the recorded data correlates well with the model. It can thus be assumed that the stacker model provides a sufficiently accurate representation of the way material is stacked onto the stockpile.

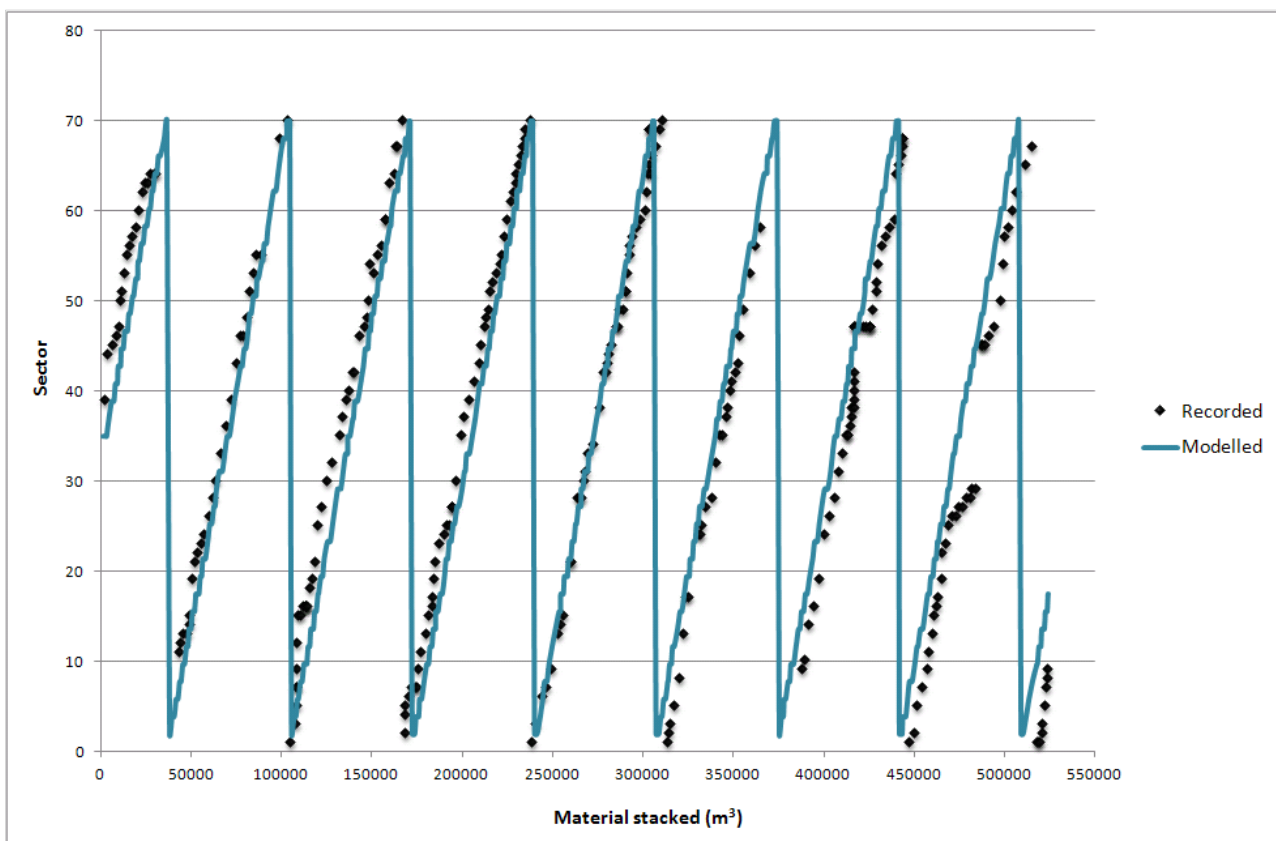


Figure 40: Accuracy of stacker geometric modelling

In Figure 41 similar results are shown for the reclaimer model. Again the simulated data follows a linear relationship between reclaimer movement and the amount of material reclaimed. A good correlation is obtained between simulated and recorded data, therefore it can be assumed that the reclaimer model provides an accurate representation of the way that material is reclaimed from the stockpile.

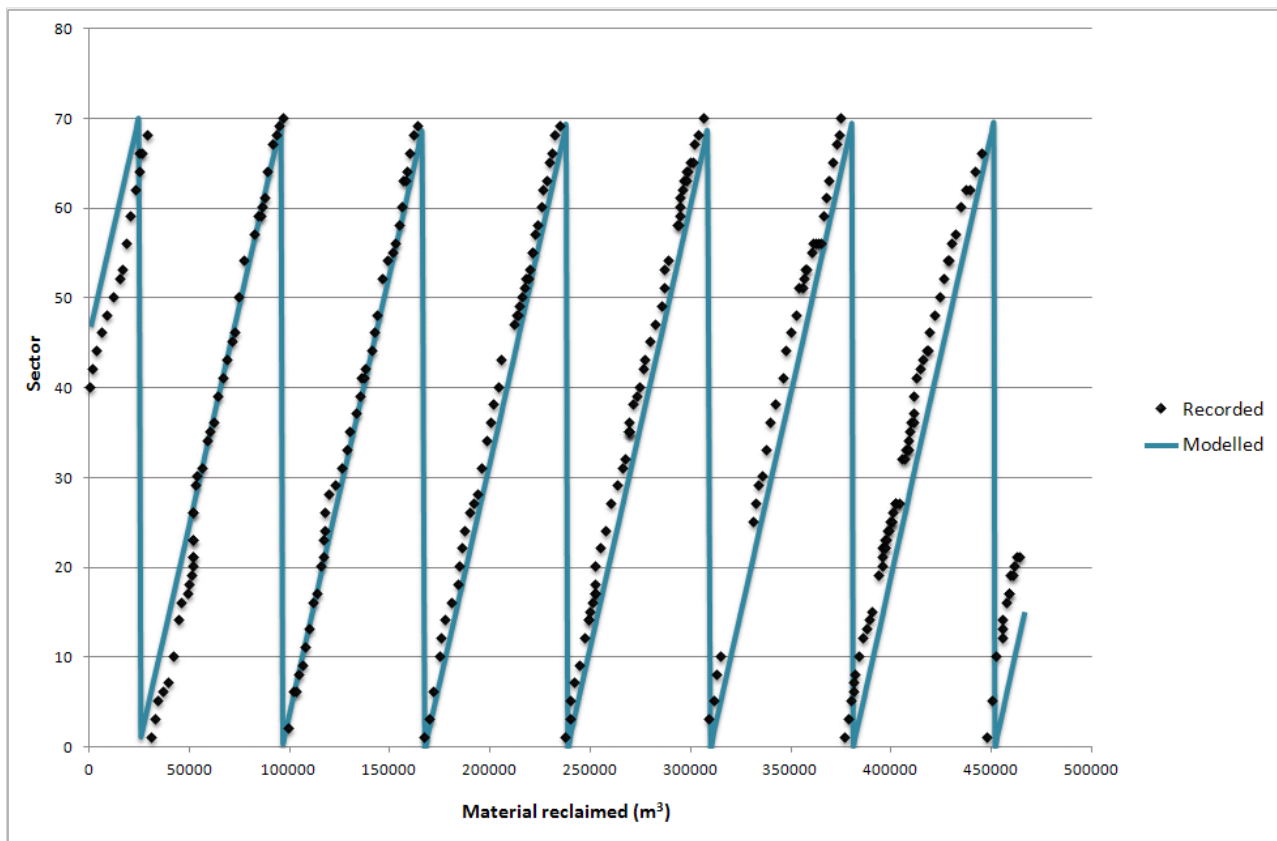


Figure 41: Accuracy of reclaimer geometric modelling

### 5.1.2. Accuracy of grade forecast

Figure 42 shows the grade of material reclaimed as predicted by the model plotted versus actual grades for the same time period. Recorded data is plotted as raw points and also as a moving average (period 4) smoothed curve. The difficulty with this validation is that the recorded data is from samples taken on a conveyor that receives feed from more than one source. Although the reclaimer conveyor is the primary feed source, it is sometimes necessary to blend in material from other sources. Therefore, although a certain grade of material is reclaimed from the stockpile, it might be blended with material of higher or lower grade before reaching the sample point. Grades and/or tonnages are also outstanding for certain data points. Since the simulation model was built on the assumption that data is available at regular intervals, values needed to be interpolated or estimated for these missing points. These approximations will generally be sufficiently accurate, but deviations between modelled and recorded data can be induced by the process of filling in points.

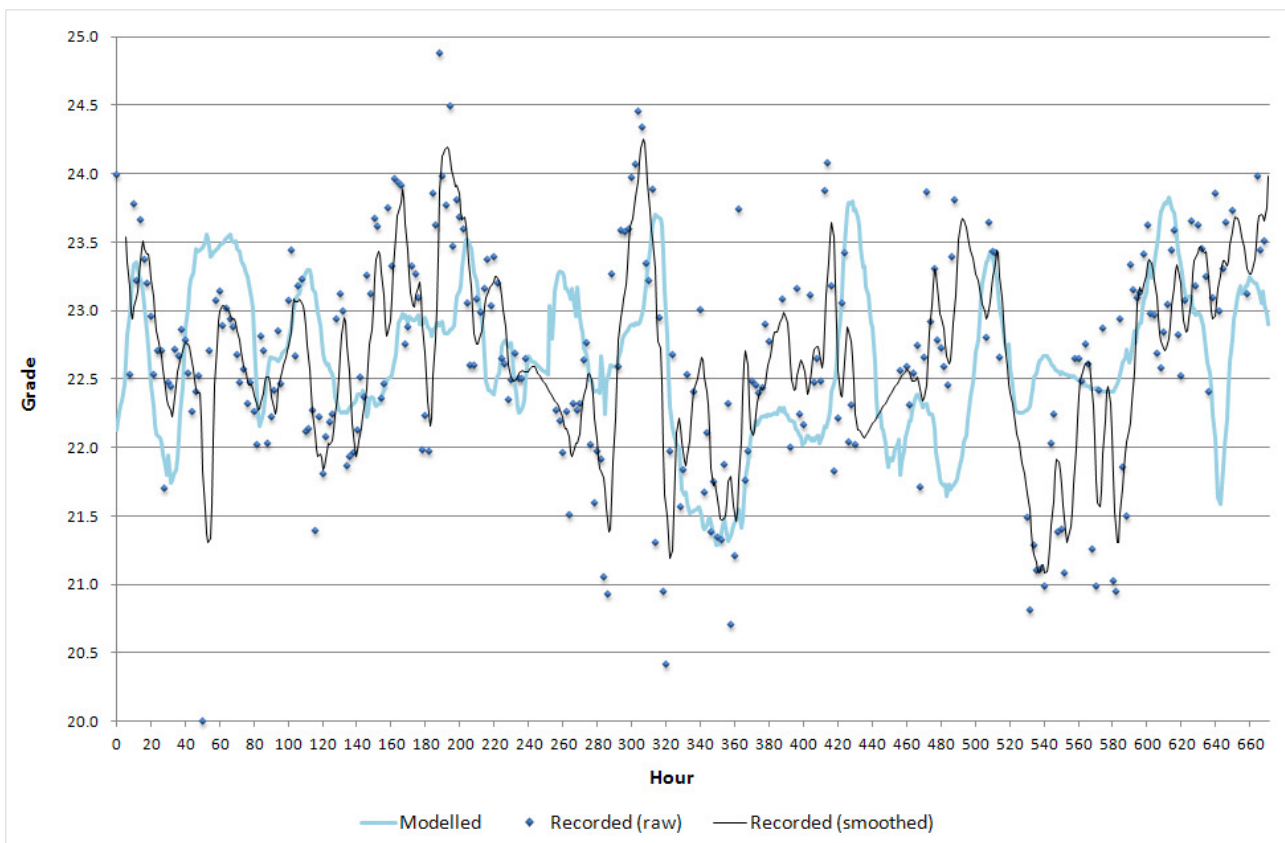


Figure 42: Coneshell model - Accuracy of grade<sup>6</sup> forecast

Table 7 shows a summary of statistical parameters for the data set which was evaluated. The first important point to note is the difference in mean value between the stacked grades and the recorded output. A stockpile is not a processing or separation operation; it cannot change the overall grade of material, only its distribution. Therefore the mean grade of material stacked to the stockpile should equal the mean grade of material reclaimed from the pile. The higher recorded mean grade shown in Table 7 is indicative of a higher grade of material being added to the reclaimer output flow. The practice of intermittently adding material from other sources also induces variation in grade values, which would explain why the recorded variance is so much higher than the modelled value.

Table 7: Coneshell validation - Statistical parameters for modelled and recorded data

Coneshell model validation				
	Minimum	Maximum	Mean	Variance
Stacked	18.42	27.95	22.65	0.831
Reclaimed (recorded)	20.00	25.36	22.75	0.612
Reclaimed (modelled)	21.29	23.83	22.65	0.314

<sup>6</sup> Grades are plotted and discussed as dimensionless numbers. The actual property represented by the grade is not important, as this study is only concerned with the consistency thereof.

In order to account for the addition of other feed sources to the reclaimed material, the modelled data shown in Figure 42 was adjusted according to the quantity and grade of other material present. For some of these other sources the grade values were known, but for others values needed to be estimated according to the type of material added. The adjusted data is shown in Figure 43.

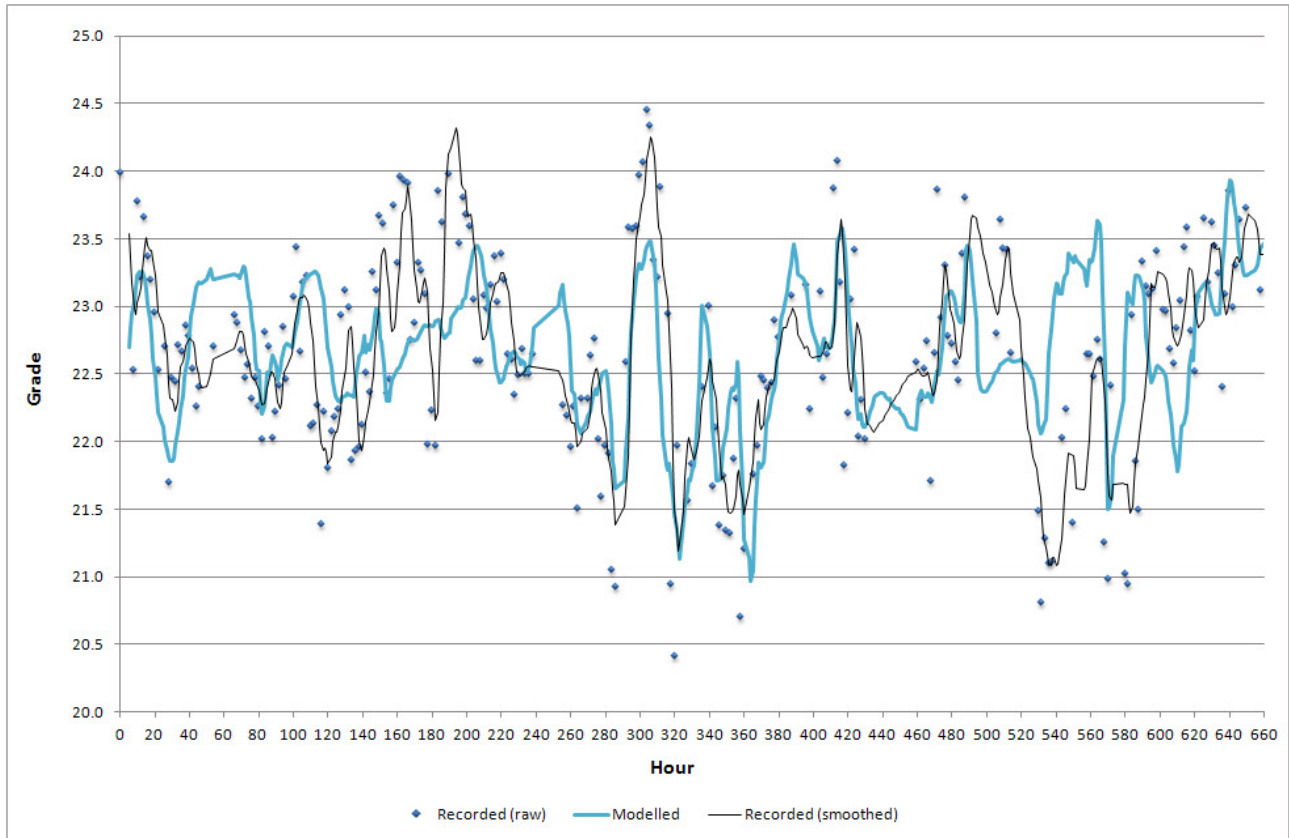


Figure 43: Coneshell model - Accuracy of adjusted grade forecast

The modelled data in Figure 43 shows higher correlation with the recorded data, although deviations are still present. The estimation of grades for the other feed sources involves much uncertainty, and it is not possible to say what the exact effect of these inaccuracies would be on the model correlation. In order to assess the effectiveness of the adjustments made, the process was repeated for an additional data set from August 2012.

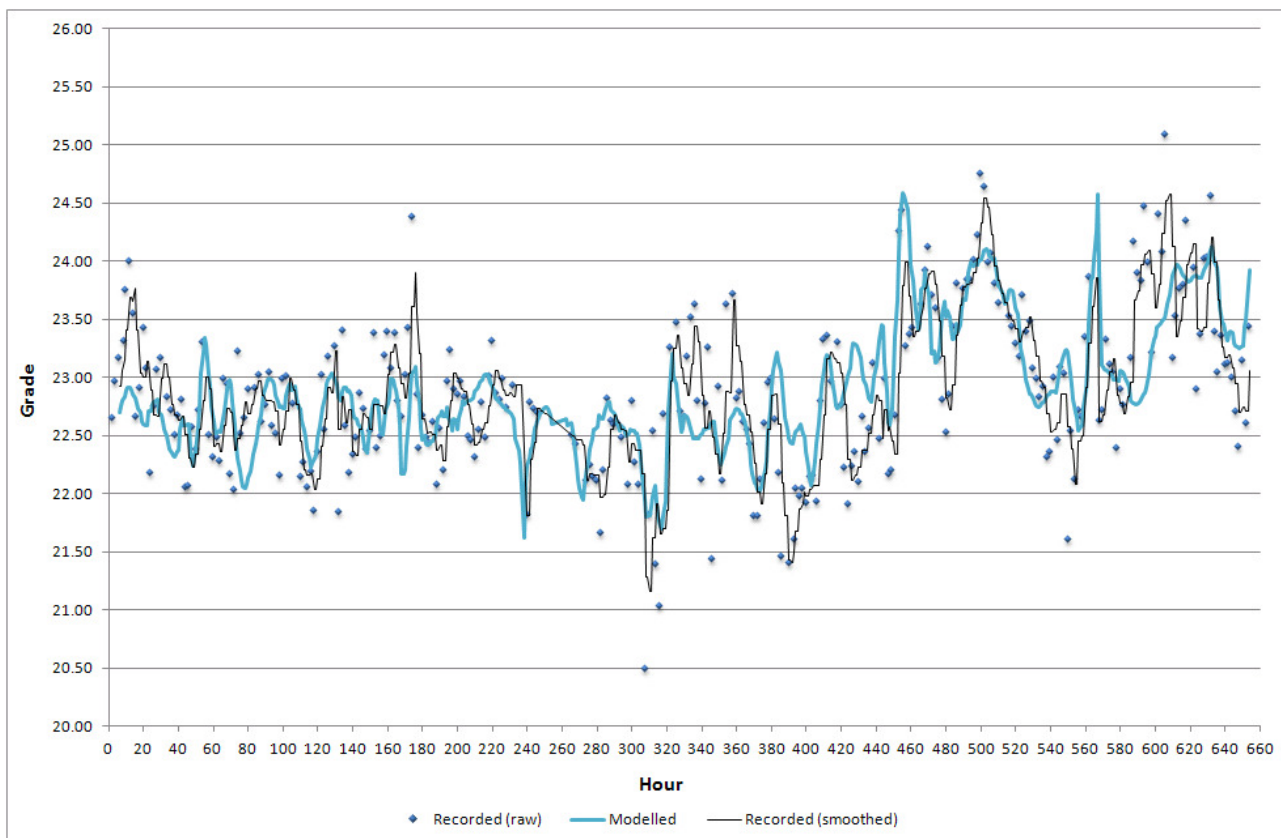


Figure 44: Coneshell stacking - Accuracy of additional adjusted grade forecast

In Figure 44 a good correlation between modelled and recorded data is observed. This could be ascribed to a number of factors, including differences in the type and quantity of additional material added to the reclaimed product for each of the months. Statistical analysis was performed on the data sets from December 2011 and August 2012 in order to determine the degree of correlation between modelled and recorded values. Deviations were calculated as a percentage of the recorded value, and it was found that for both data sets the deviations followed a normal distribution centered around 0% deviation. Therefore the modelled values had roughly equal probability of being either higher than or lower than the corresponding recorded data. Additional information gathered during the statistical analysis is shown in Table 8.

Table 8: Coneshell stacking - Correlation between modelled and recorded data

Deviations between modelled and recorded data						
	Minimum	Maximum	Mean	Variance	$\mu - 2\sigma$	$\mu + 2\sigma$
December 2011	-8.80%	6.22%	-0.01%	1.98%	-3.96%	3.94%
August 2012	-5.34%	4.39%	-0.16%	1.38%	-2.76%	2.60%



Table 8 shows that the deviations between modelled and recorded data were much smaller for the August data set. However, even for the seemingly poor correlation of the December data, 95% of modelled data still fell within 4% of the recorded value. These results suggest that the stockpile model can provide a reasonably accurate prediction of peaks and troughs in reclaimed grade.

## 5.2. Chevcon model

In order to account for limitations in validation of the chevcon model, three alternative validation measures were used. The first two address the accuracy of the geometric modelling of a chevcon pile, while the third seeks to evaluate if the grade predictions as generated by the chevcon model are reasonable.

1. **Mass balance.** The amount of material reclaimed from the pile should equal the amount of material stacked to the pile. Any deviation between these two numbers would be an indication of inaccuracies in the geometrical modelling.
2. **Average grade.** As noted previously in section 5.1.2, the average grade of material stacked to the pile should equal the average grade of material reclaimed. This property can, to some extent, be used to evaluate the integrity of grade predictions.
3. **Analysis of grade predictions.** The chevcon model is expected to smooth the grade variations present in the stockpile input. By studying the output variations, and matching them to the input, a partial validation of the accuracy of grade predictions can be done.

The chevcon model was used to simulate stacking and reclaiming the circular blending pile over a period of one month, as for the coneshell model. Input grades and tonnages were taken from the December 2011 dataset. Subsequently the accuracy of the simulation results was evaluated at the hand of the three measures introduced above.

### 5.2.1. Mass balance

In the interest of comparison, the validation measures used for the chevcon model were also applied to the two coneshell validation simulations. Results of this comparison are shown in Table 9.

**Table 9: Mass balance over validation simulations**

	Stacked tonnes	Reclaimed tonnes	Deviation
Chevcon model - Dec	477419	461085	3.4%
Coneshell model - Aug	471684	450450	4.5%
Coneshell model - Dec	470602	465802	1.0%



In all three the simulations shown the mass balance was accurate to within a range of 5%. Moderate deviations between stacked and reclaimed masses can be expected, resulting from the geometrical approximations used to develop the models. From the results in Table 9 it can be concluded that, based on this measure alone, the geometric modelling used for chevcon stacking is as accurate as that for coneshell stacking.

### 5.2.2. Average grade

In the same way as was done in the previous section, the coneshell model results were used to investigate whether results generated by the chevcon model are plausible. Table 10 shows the average grade of the stacked material and of the simulated output material for each of the validation simulations.

**Table 10: Average grade stacked and reclaimed for validation simulations**

	Average grade stacked	Average grade reclaimed	Deviation
Chevcon model – Dec	22.71	22.69	<0.1%
Coneshell model – Aug	22.97	22.95	<0.1%
Coneshell model – Dec	22.71	22.71	0.0%

Average grades for all simulations show good correlation between the input and output streams. This result indicates that none of the models produce biased results by for instance “throwing away” some of the stacked grades. It would therefore be reasonable to assume that all grades that are stacked are reflected in the output streams of the respective stockpile models.

### 5.2.3. Analysis of grade predictions

As noted in section 2.1.2, material property fluctuations in the feed to a blending pile are expected to be smoothed in the output. Therefore the accuracy of the chevcon model grade predictions can be evaluated by means of inspection. In Figure 45 the predicted variation of reclaimed grades is compared to stacker data. An additional series, where stacked grades are smoothed over a 20 hour period, is also shown. The modelled values correlate very well with the smoothed series, as would be predicted by blending theory. This can be taken as a fair indication that the chevcon model is able to provide information about grade variation in the output of a circular blending pile.

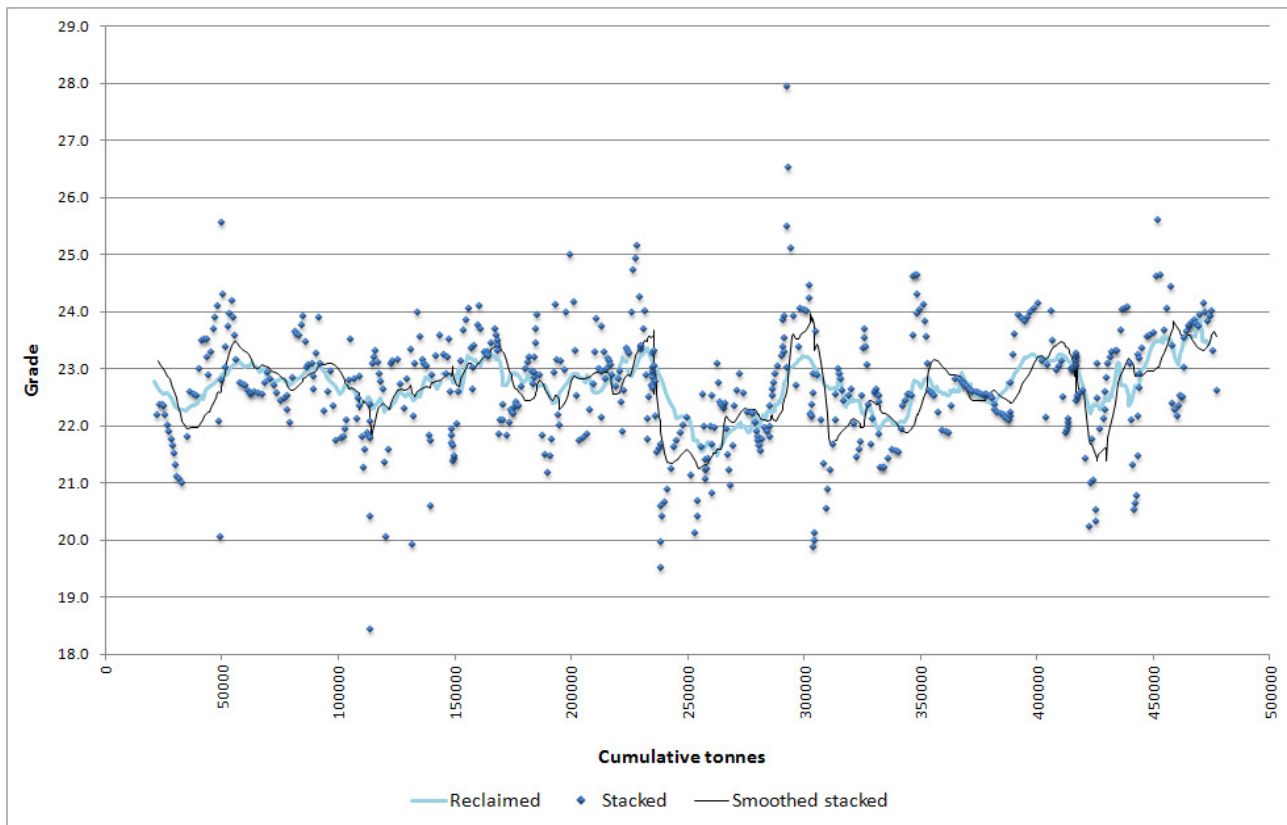


Figure 45: Chevcon model - Accuracy of grade forecast

Statistical data for each of the three series shown in Figure 45 is provided in Table 11 below. This preliminary validation of the chevcon simulation model has provided some insight into the accuracy of model development, the results of which would need to be confirmed when actual output data is available.

Table 11: Chevcon validation - Statistical parameters for modelled and recorded data

Chevcon model validation				
	Minimum	Maximum	Mean	Variance
Stacked (raw)	18.42	27.95	22.71	0.632
Stacked (smoothed)	21.26	23.97	22.67	0.326
Reclaimed	21.47	23.94	22.69	0.176

## 6. Sensitivity analysis

In chapter 6 the influence of different operating properties on blending efficiency is investigated by using VRR as a measure. The results serve as indication of possible adjustments that can be made to improve the consistency of output grade. This chapter also demonstrates the model's capability of predicting the consequences of decisions (i.e. to increase the stacker speed), as stipulated in the design criteria.

### 6.1. Coneshell stacking

As mentioned in section 3.4, two parameters were evaluated in the coneshell stacking model. The results are discussed below.

#### 6.1.1. Stacker step increment

Figure 46 shows how VRR varies in relation to the stacker movement increment. As the distance the stacker moves between cones is increased the VRR can be seen to increase almost linearly. This is an expected result, as smaller increments between cones is analogue to more layers being stacked in any given space, which would result in better blending efficiency. The increment between cones should thus be minimized as far as possible. Another important result from Figure 46 is that as the stacker increment tends to zero, the VRR tends to 0.25. Therefore, with all other stockpile parameters as is and in input variance of 0.831, a minimum variance of 0.208 can be achieved with coneshell stacking. According to the empirical rule of statistics (see section 2.1.5) it means that if material with an average grade of 22.65 is stacked 95% of the output grades will fall between 21.74 and 23.56.

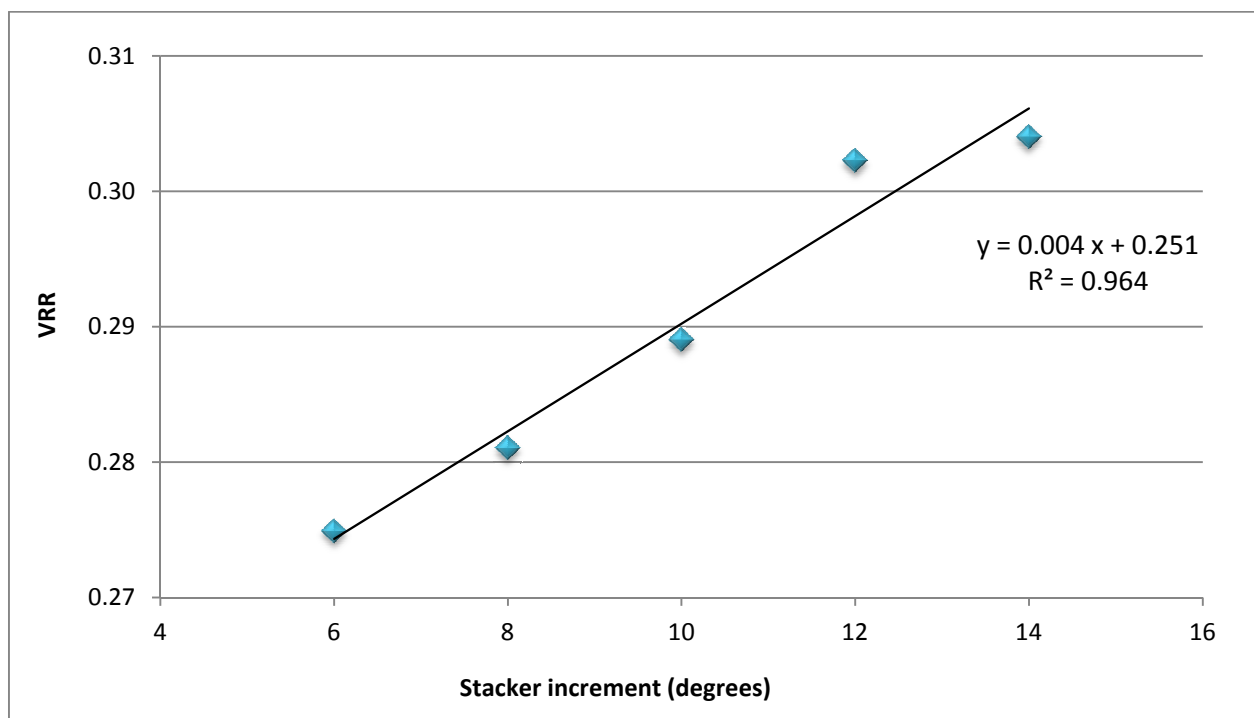


Figure 46: VRR plotted as a function of stacker increment

### 6.1.2. Stacked height

Figure 47 shows how VRR varies with stacked height. Although the VRR data points resemble a straight line, a linear relationship is not possible. If VRR were to decrease linearly with an increase in stacked height it would mean that at a point before 25m stacked height the line would cross the x-axis, implying zero variance. Since completely consistent output from a blending pile is not practically achievable it makes more sense to approximate the relationship between VRR and stacked height as an exponential function.

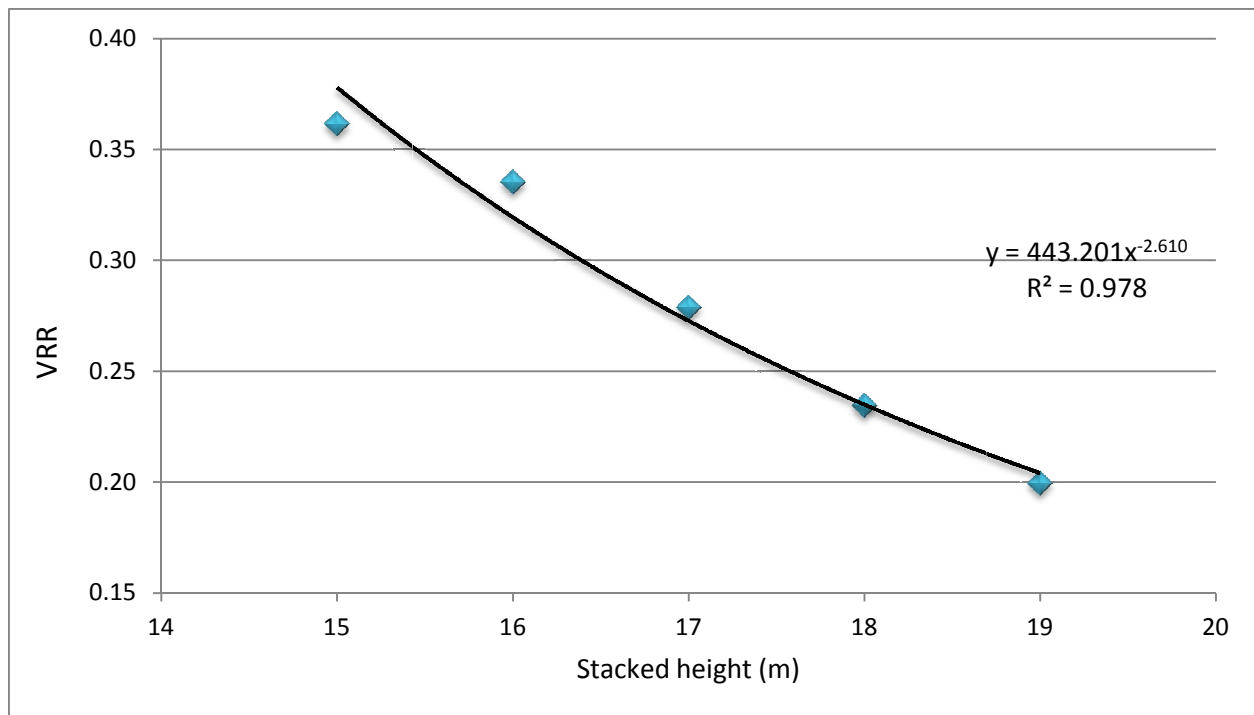


Figure 47: VRR plotted as a function of stacked height

An increase in maximum height of the stacked pile will rarely be possible, since the diameter of the stockpile would have to increase accordingly. Figure 47 does however shed light on the impact of not stacking a stockpile to its full capacity, as observed at Colliery X. Best practice would be to always keep the stacker boom lifted to its maximum stacking height.

## 6.2. Chevcon stacking

The influence of three different variables on the expected output variance was investigated. The results are displayed and discussed below.

### 6.2.1. Rotation angle

The angle that the stacker rotates with every new chevcon layer stacked can also be expressed as blending tail length. Gerstel (1989) stated that in order for sufficient blending to take place on a chevcon-stacked circular blending pile the tail length had to be at least four times the horizontal length of the reclaiming harrow, i.e. the distance between points A and B in Figure 34.

This can be calculated as:

$$reclaimerLength = \frac{hStockpile}{\tan(rakeAngle)}$$

Equation 41: Calculation of reclaimer length

Figure 48 demonstrates how output variance decreases with increasing blending tail length. The relationship between VRR and tail length can also be approximated as an exponential function. According to Gerstel's rule of thumb the blending tail for the stockpile in question would need to be at least 87m, but Figure 48 would suggest that increasing the blending tail beyond 60m would not yield significant further decrease in variance. When evaluating this parameter one must also keep in mind that increasing the size of the blending section decreases the stockpile storage capacity. Incremental improvements in grade consistency must thus be weighed against buffering needs.

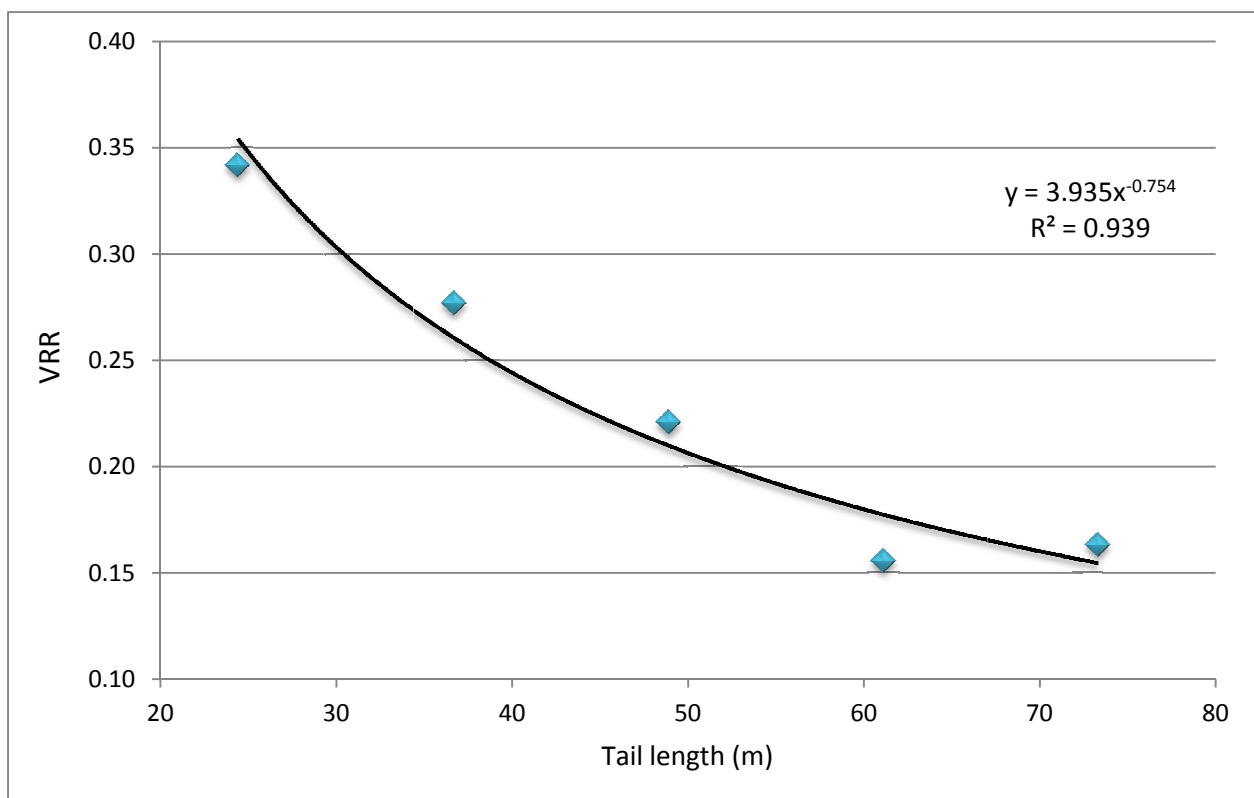


Figure 48: VRR plotted as a function of blending tail length

### 6.2.2. Stacker speed

From the work of Bond et al (2000) an increase in stacker speed is expected to yield a decrease in output variance, as faster stacker movement means that material from each time increment gets stacked into more layers. Figure 49 shows that when VRR is plotted as a function of stacker speed however, the data can be fit to a parabolic curve. Although this would imply that values of stacker speed exist for which variance equals zero, which has been established as a practical impossibility, the parabolic shape seems a viable

result for this section of the graph. The curve suggests that for low stacker speeds there is little difference in the resultant variance, but there exists a critical speed after which variance decreases dramatically. From this result it can be concluded that, as far as is practically viable, stacker speed should be maximized if a consistent output grade is desired.

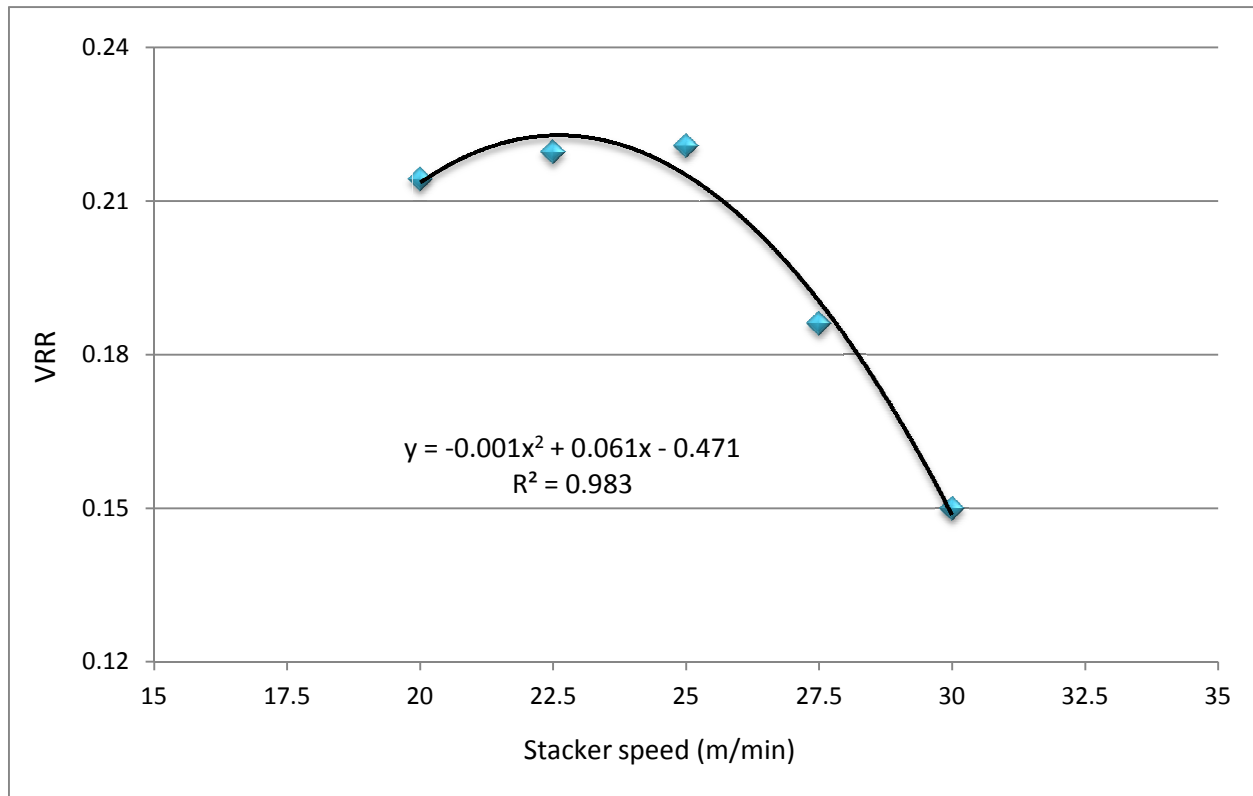


Figure 49: VRR plotted as a function of stacker speed

### 6.2.3. Number of layers / step increment

The number of chevcon layers which contact the reclaimers face was related to the stacker step increment in Equation 30. The number of layers was used to specify the simulation parameters, but exchanged for step increment in discussion of the results, as this is a more relatable parameter. Figure 50 shows how VRR changes with increasing stacker movement increment between successive chevcon layers being stacked. The graph doesn't seem to follow a recognizable form, but some conclusions can be drawn. It appears that for very low values of the stacker increment (smaller than one degree) the model delivers inaccurate results. This could be because the layer increment is not sufficiently large in relation to the calculation increments used by the stacker and the reclaimers, which would mean that some of the approximations used are now insufficient. These inaccuracies induce false variance in the results. A solution to the problem would be to use smaller calculation increments to model the behaviour of the stockpile for very small values of layer increment. For values of layer increment greater than and equal to one degree the relationship to VRR has a more expected form: variance is reduced if smaller increments of stacker movement are used (as concluded for the coneshell model in section 6.1.1). For this situation it appears

that the optimal value of layer increment is one degree, but the result would need to be investigated further by running the model on a PC with more processing power.

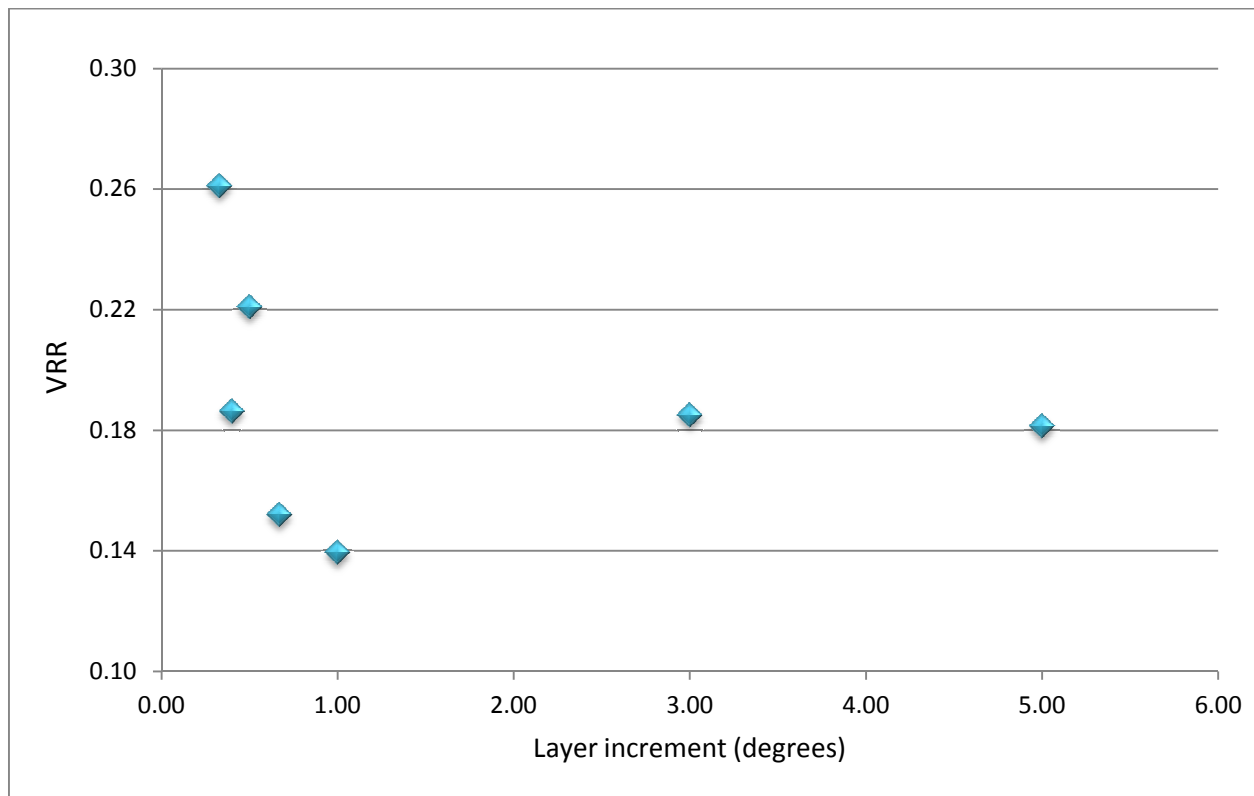


Figure 50: VRR plotted as a function of layer increment

### 6.3. Coneshell vs. Chevcon stacking

In order to compare the output variance generated by the coneshell and chevcon methods for the same input, one additional simulation was run for each model. These simulations used the optimal values of each of the parameters as identified in sections 6.1 and 6.2. For the coneshell model a stacker movement increment of six degrees was combined with a stacked height of 19 meters. In the chevcon simulation the stacker rotated about an angle of 150 degrees for every layer, stacking speed was set to 30m/min, and a layer increment of one degree was used. The results are shown in Table 12.

Table 12: Comparison of coneshell and chevcon stacking

	Mean	Variance	VRR		
Coneshell stacking	22.66	0.203	0.236	21.76	23.56
Chevcon stacking	22.71	0.075	0.121	22.16	23.26

The two models didn't use exactly the same data set, as the long blending tail of the chevcon model means that a smaller volume is stacked, which lead to some of the later data points not being included. This explains the difference in mean values and input variances observed in Table 12. For the coneshell model a VRR of 0.236 is obtained. This is not a significant reduction from most of the other coneshell stacking

results, which seems to imply that variation of the stockpile parameters has a limited influence on the blending efficiency for a stacker in coneshell mode. The result leads to the conclusion that, for the stockpile used in simulations, VRRs of less than 0.2 will unlikely be achieved while stacking in coneshell mode. For the chevcon stacking mode however a large reduction in variance is obtained by optimising the stacker parameters. The very low output variance indicates that, for the right stockpile configuration, near-perfect blending can be achieved by stacking in the chevcon mode. The result will of course be subject to induced variances, like that caused by sampling (see section 2.3.4).

From the results discussed chevcon stacking has proven to deliver much better consistency in the output grades reclaimed from a circular stockpile. As reported by Bond et al (2000) and Wintz (2011), coneshell stacking is best reserved for when homogenisation is not an important requirement. To explain a potential implication of this conclusion an example is used:

A stream of material with average grade 23 and variance of 0.7 is fed to a circular stockpile stacked in chevcon mode. If a VRR of 0.15 is assumed, the output variance is calculated as 0.105. This means that 95% of the output material will have a grade that varies between 22.35 and 23.65. The minimum and maximum values are especially important when the customer has strict regulations on minimum and/or maximum grade. If the same material is stacked in coneshell mode, assuming a VRR of 0.25, the output variance is calculated as 0.175. The output grade would now vary between 22.16 and 23.84. If a minimum grade of 22.35 were imposed, only two courses of action would guarantee compliance to the specification:

1. **Reduce input variation.** If the input variation were reduced to 0.42, a VRR of 0.25 would yield an output variance of 0.105, and grades within the specified limits.
2. **Increase the mean grade.** If the mean grade were increased to 23.2 the lower limit, with the same variance, would increase to 22.36.

Both these solutions would be very difficult to achieve. Input variation is rarely a variable that can be controlled, and with high grade material scarce as it is, an increase in mean grade could be too much to ask. Therefore an investment in chevcon stacking infrastructure and proper focus on the control of blending efficiency could prove to be a worthwhile utilisation of resources and time.



## 7. Model application

The stockpile simulation model was built as part of a decision support tool that serves to aid in the design, management, and optimisation of a circular blending pile. Model applications have been mentioned throughout this document, but the chapter to follow seeks to further highlight the model's role in management decisions. The following three areas of management are addressed:

- Operations research
- Operations management
- Financial management

### 7.1. Operations research

Operations research is the application of advanced analytical methods to facilitate better decision-making. It is a branch of applied mathematics which uses various scientific research-based principles, strategies, and analytical methods including mathematical modelling, statistics and algorithms to improve an organisation's ability to make meaningful management decisions (Institute for Operations Research and Management Sciences, 2012).

The development of the stockpile simulation model combined application-specific research, mathematical relations, and problem-solving strategies to generate a decision support tool. This tool provides the operations manager with the right information needed to make effective decisions for the benefit of the business. The stockpile management model is therefore an example of how operations research can benefit industry, going beyond theory to provide practical solutions to the problems present in a production environment.

### 7.2. Operations management

Operations management can be defined as the management of resources used to create products and services. It is concerned with the activities that are necessary to turn business inputs into more valuable business outputs (Finch, 2008). Therefore the aim of operations management is to control the production process in such a way that customer specifications are met, while minimising the resources needed.

Specific operations management applications of the decision support tool involve day-to-day production planning. The first question mentioned in section 1.2 is used as an example, repeated here:

- "Taking into account the material that is currently on the pile, will adjustments to the downstream process be necessary?"

Without use of the simulation model the decision making process would typically involve large amounts of estimation and “educated guesses”, as shown in Figure 51 below. The production/operations manager starts the day by planning for any changes that needs to be made to the production process. For the downstream operation that uses feed from the stockpile as an input, the material grade from the stockpile is the primary consideration in production planning. When changes to the production process involve the sourcing of different raw materials, as an example, it is important to communicate with suppliers in due time. Any production risks, like not being able to adhere to product specifications, also need to be identified during this planning process.

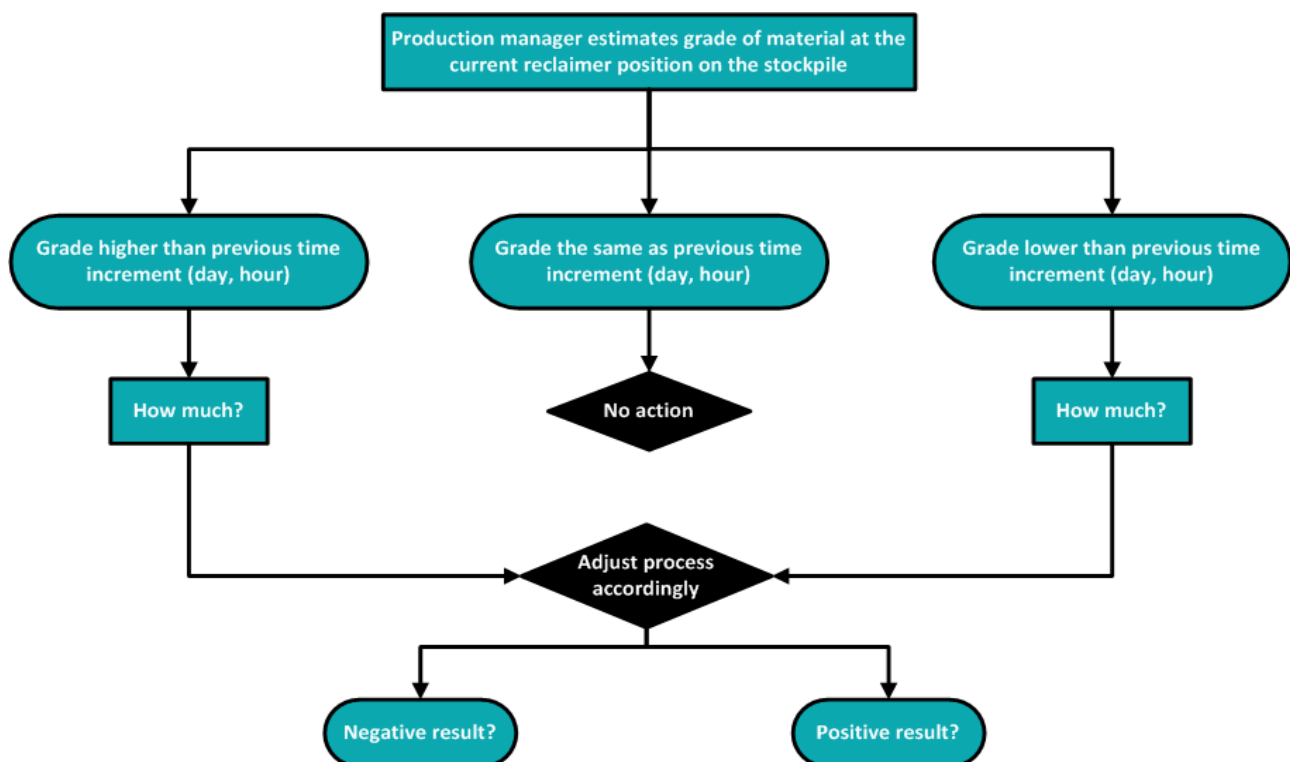


Figure 51: Simple production planning decision making process

The final outcome of the decision process, namely the positive or negative consequence, is directly related to the first step in the process: estimating the stockpile output grade. The estimation is, in many cases, based on nothing more than rules of thumb. By implementing the decision support tool this step is replaced by “model the stockpile output grade”, adding a degree of scientific rigour to the decision making process. Even if the model forecast proves to be only 95% accurate (based on validation simulations in Chapter 5), it would still be more likely to result in the desired consequence.

Alter (1977) raises the concern that representational models often have problems with credibility. This will be especially true in the intended mineral processing environment where the decision support tool would be utilised. Typical users of the tool would be highly experienced in their field, and therefore apprehensive about accepting the information generated as reliable. It is therefore crucial that the model’s validity be

constantly proved to the users thereof. This can be done by relating the decision consequence back to the model information used, and enforcing confidence in the relationship between the two. For the purpose of this validation it matters not whether the decision consequence is positive or negative, but whether, in light of the model predictions, it is **expected**.

As was noted in section 2.2, the gathering of information is a vital part of an effective decision making process. The stockpile management model can assist the operations manager in making decisions about production scheduling, material allocation, and operational parameters by providing the necessary information. Depending on the stockpile capacity and the depletion rate, a forecast can be generated which provides the grade information anything from a few hours to a few days in advance. By making use of a DSS an element of planning can be introduced that would not be present otherwise.

Furthermore, without the simulation model available, operational parameters would be optimised on a trial-and-error basis in order to maximise blending efficiency. This can be a lengthy and sometimes unsuccessful exercise, destroying value in the process. By using simulation to establish the influence of operational parameters like stacker speed and movement increment, instead of making changes on an intuitive or haphazard basis, optimal values can be found quicker and easier. This also ensures that the focus remains with the parameters that have been identified to contribute the most towards the blending efficiency goal.

### 7.3. Financial management

The stockpile simulation model can contribute to the financial management function of a production facility in two ways. Firstly, the production manager needs to be concerned with minimising costs and maximising revenue on a daily basis. By providing the customer with a product of the right specification while closely monitoring the use of raw materials this objective can be achieved. If attention is paid to the way that raw material and product stockpile grades are managed, as can be facilitated by the stockpile simulation model, a more consistent and efficient process can be maintained. Some customers provide financial incentives in the form of “consistency bonuses”, or charge penalties for a product that does not meet their specifications. These bonuses and/or penalties can have a marked effect on the operational profit generated.

Secondly, the stockpile model can aid in the decision-making process when capital expenditures are considered. If, for instance, a change from coneshell to chevcon stacking is planned, a new stacker drive motor would need to be purchased and installed. An expansion of stockpile capacity or change of reclaimers type will also result in huge capital expense. These expenses need to be weighed against the possible advantages that will result from execution of the relevant capital projects. The stockpile management model provides a way to quantify the improvement in blending efficiency, and calculate resultant financial gains.

## 8. Conclusions

The objective of this study was to develop a decision support tool which will enable the user thereof to effectively manage a circular stockpile. The tool developed was used to answer the research question; whether consistency of material grade, as reclaimed from a circular stockpile, can be predicted.

The decision support tool developed was classified as a representational model-driven PDSS, aimed at supporting decisions by providing information. At the core of the tool is a model that simulates the behaviour of a continuous circular blending pile. The model aimed to serve as a sufficiently accurate representation of stockpile behaviour, which would be able to generate data pertaining to the variation of output material grade. Development of the simulation model extended work done by other authors by eliminating the assumptions of constant stacker flow and vertical reclaimer slices.

The objectives set out in section 1.3 were achieved by developing simulation models for both chevcon and coneshell stacking by using the VBA platform, as discussed in Chapter 4. Variable feed rate is accommodated, and reclamation is modelled by using an inclined reclaimer rake. A user interface was developed that makes the complex calculations behind the simulation easily accessible to operating and management personnel. It is important to emphasize that the simulation model serves as a decision *support* tool only. This means that the model does not control the process in any way, or even suggest a course of action in decision-making. It merely presents information in a way that will supplement the knowledge and experience of the user.

The validity of the simulation model was evaluated by using data from a case study. Information about the material fed to the circular stockpile was used to make predictions about the stockpile behaviour. It was found that the coneshell model offers a reasonable representation of reality in the way that it simulates the movement of the stacker and reclaimer units around the pile. This result suggests that the accuracy of geometric modelling was sufficient. The grade of material reclaimed from the blending pile was predicted by the coneshell model and compared to actual sample data, and modelled values fell within 4% of the recorded grades. Despite challenges to model validation as a result of incomplete data and other factors, the simulation model proved able to predict peaks and troughs in the output grade. If a production manager had this forecast available they would thus be able to plan proactively instead of operating reactively when low grade material is reclaimed.

In order to account for data availability limitations in the validation of the chevcon model, three alternative measures were used to assess the validity of model outputs. A mass balance was performed over the limits of the pile, in order to confirm that the amount of material reclaimed from the pile, as reported by the simulation model, equals the material stacked to the pile. The mass balance produced a positive result, indicating that the geometric modelling of the pile is within acceptable limits of accuracy. The average

material grade in the input and output of the pile was compared, and found to differ with  $<0.1\%$ . This result served as an indication of the integrity of grade predictions generated by the chevcon model. Finally the grade predictions were analysed, matching it to a smoothed version of the input variations. The evaluation correlated well to the principles of bed blending theory. The preliminary validation of the chevcon simulation model provided some insight into the accuracy of model development, the results of which would need to be confirmed when output data from a chevcon-stacked pile is available.

One of the significant advantages of using a simulation model is that the effect of changes can be investigated without actual implementation, and potentially negative outcomes can be avoided. This enables the user to optimise their process through risk-free experimentation. In Chapter 6 the simulation models were used to investigate the sensitivity of blending efficiency to various input parameters. It was found that for coneshell stacking a smaller increment between stacked cones will result in lower output variance, with a close to linear relationship being observed. The results also suggested that a minimum output variance exists, which corresponds to a VRR of approximately 0.25 for the range of parameters investigated. The influence of stacked height on blending efficiency was also evaluated for the coneshell stacking model. It was found that in order to minimise output variance the stacked height must be maximised. The impact that this operating variable can have will be limited, as the stacking height cannot be arbitrarily increased. In order to facilitate an increase in stockpile height a new stockpile must be designed. Both of these results confirm findings from literature. Denny & Harper (1962), De Wet (1983, 1994), and Petersen (2004) all noted that increased blending efficiency can be obtained by stacking more layers or increasing the stockpile capacity. Stacking more layers in coneshell mode is analogue to decreasing the distance between cones, while an increase in stacked height would result in an increase in stockpile capacity.

For a chevcon-stacked pile a decrease in output variance is observed for longer blending tails. The results suggest however that beyond a blending tail angle of about  $125^\circ$  no significant decrease in VRR will be obtained. It is important to weigh the increased blending efficiency against the decrease in available stockpile volume when considering this factor. A rotation angle of  $125^\circ$  means that more than a third of the stockpile volume will not be available as buffer capacity. Simulation results for varying stacker speeds indicate that there exists a critical speed of about 25m/min above which a large reduction in output variance can be achieved. Therefore the stacker speed should be maximised as far as is practically viable. The last parameter that was evaluated is the increment that the stacker moves between chevcon layers. For very small values of the movement increment, smaller than 1 degree, the model proves to be inaccurate. This could be because the layer increment is not sufficiently large in relation to the calculation increments used by the stacker and reclaimer, which means that some of the approximations used would prove to be insufficient. Beyond one degree an increase in layer increment results in an increase in output variance.

In order to compare the output variance generated by the coneshell and chevcon stacking methods for the same input, additional simulations were run for each model. The simulations used optimal values of stockpile parameters as identified in the sensitivity analysis. It was found that for coneshell stacking optimisation of the input parameters yields little improvement in blending efficiency, as the reduction in output variation that can be obtained is limited. It was concluded that VRRS of less than 0.2 would unlikely be achieved while stacking in coneshell mode. This is an expected result, as findings from Bond (2000), Erasmus (2001), and Wintz (2011) all report high output variation for the coneshell stacking mode, deeming it unsuitable for applications requiring high blending efficiency. The output variance generated by the chevcon model was half of that achieved with coneshell stacking, a VRR of less than 0.1, proving that the investment in chevcon stacking infrastructure is one that every production facility seeking improvement in output grade consistency should consider. This also confirms Bond's approximation of a 10:1 reduction in variance for a stockpile stacked in chevcon mode and reclaimed with a full-face reclaimer (Bond, 2000).

The research objectives set at the start of the study have been achieved, and results indicate that the decision support tool can be used to predict and control the consistency of material grade as reclaimed from a circular blending pile. Chapter 7 notes how the simulation model contributes to different elements of management, discussing its role in operations management, financial management, and operations research. The decision support tool will be put through its first true test during implementation at Colliery X, which will establish a base for further studies.

## References

- Alter, S. 1977. *A Taxonomy of Decision Support Systems*. Sloan Management Review, 39.
- Arnott, D., & Pervan, G. 2005. *A critical analysis of decision support systems research*. Journal of Information Technology, 20(2), 67-87.
- Bohanec, M., & Rajkovic, V. 1987. *An expert system approach to multi-attribute decision making*. In Proceedings of IASTED International Conference "Expert Systems" (ed. MH Hamza), Acta Press.
- Bohanec, M. 2003. *Decision Support*. Data Mining and Decision Support, 23-35.
- Bond, J.E., Coursaux, R. & Worthington, R.L. 2000. *Blending systems and control technologies for cement raw materials*. Industry Applications Magazine, IEEE, 6(6): 49-59.
- Carpio, R.C., Coelho, L., Silva, R. and Jorge, A. 2005. *Case Study in Cement Kilns Alternative Secondary Fuels Mixing using Sequential Quadratic Programming, Genetic Algorithms, and Differential Evolution*. Paper presented at Proceedings of 6th World Congress on Structural and Multidisciplinary Optimization, Rio de Janeiro, RJ, Brazil.
- Clemen, R. T. 1996. *Making hard decisions: an introduction to decision analysis*.
- De Wet, N. 1994. *Homogenizing/Blending in South Africa - an update*. Bulk Solids Handling, 14(1): 93.
- De Wet, N. 1983. *Homogenizing/Blending plant applications in South Africa with special reference to Gencor's Hlobane and Optimum plants*. Bulk Solids Handling, 3(1): 55.
- Denny, R.J. and Harper, W.G. 1962. *The Homogenisation of Raw Coal*. Paper presented at Fourth International Coal Preparation Congress.
- Dictionary.com. 2012. Consistency. [Online].  
Available: <http://dictionary.reference.com/browse/consistency> [27 June 2012]
- Efstathiou, J., & Rajkovic, V. 1979. *Multi-attribute decisionmaking using a fuzzy heuristic approach*. Systems, Man and Cybernetics, IEEE Transactions on, 9(6), 326-333.
- Erasmus, J.H. 2001. *Bulk Raw Materials Storage Selection*. Paper presented at Beltcon - South African Institute of Materials Handling.
- Everett, J.E. and Kamperman, M. 1999. *A Decision Support System to Aid Quality Control in Iron Ore Production*. Paper presented at International Conference on Industrial Logistics. St Petersburg, Russia. 28 June to 3 July.

- Everett, J. 2007. *Computer aids for production systems management in iron ore mining*. International Journal of Production Economics, 110(1-2): 213-223.
- Finch, B. J. 2008. *Operations Now*. McGraw-Hill
- Gerstel, A.W. 1989. *Blending in continuous piles*. Bulk Solids Handling, 9(4): 413.
- Gerstel, A.W. 1979. *The homogenisation of bulk material in blending piles*. Unpublished thesis. Delft University of Technology.
- Gilfillan, L. 1997. *Project Management and Evaluation*. [Online].  
Available: <http://lga-inc.com/ut/syllabus/Session7and8/index.htm>
- Gupta, A., Das, A. & Chauhan, G. 2007. *A coal-blending model: A tool for better coal blend preparation*. Coal Preparation, 27(1-3): 28-38.
- Gy, P. 1992. *Sampling of heterogeneous and dynamic material systems: Theories of heterogeneity, sampling, and homogenizing*. Elsevier Science Ltd
- Gy, P.M. 1981. *A new theory of bed-blending derived from the theory of sampling — development and full-scale experimental check*. International Journal of Mineral Processing, 8(3): 201-238.
- Hillier, F. S., Lieberman, G. J., & Hillier, M. 1990. *Introduction to operations research* (Vol. 6). New York: McGraw-Hill.
- Holderbank. n.d. *Preblending of raw materials*. Paper presented at Cement seminar - Process technology.
- Hurwitz, M.J. & Ackermann, R.E. 1999. *Real-Time Optimization for Mix Beds*
- IMOS, Inc. 1997. *Decision Support Primer*. [Online]. Available: <http://www.imos.com/whatis.htm>
- INSEAD: Decision Sciences. 2001. *PhD Program Description*. [Online].  
Available: <http://www.insead.fr/phd/decisions.htm>
- Institute for Operational Research and Management Science. 2012. *About Operations Research*. [Online].  
Available: <http://www.informs.org/About-INFORMS/About-Operations-Research> [22 August 2012]
- Kamperman, M., Howard, T. & Everett, J. 2003. *Controlling product quality at high production rates as applied to BHP Billiton Iron Ore Yandi Fines Operation*. Mineral Processing and Extractive Metallurgy, 112(1): 13-18.
- Keen, P. G., & Morton, M. S. S. 1978. *Decision support systems: an organizational perspective* (Vol. 35). Reading, MA: Addison-Wesley.



- Keeney, R. L. 1982. *Decision analysis: an overview*. Operations Research, 30(5), 803-838.
- Kumral, M. 2006. *Bed blending design incorporating multiple regression modelling and genetic algorithms*. Journal of the South African Institute of Mining and Metallurgy, 106(3): 229-236.
- Kumral, M. 2005. *Quadratic programming for the multivariable pre-homogenization and blending problem*. Journal of South African Institute of Mining and Metallurgy, 105(5): 317-322.
- Lemke, K. 1962. *The Homogenisation of Raw Coal Properties: Techniques Costs and Benefits*. Paper presented at Fourth International Coal Preparation Congress.
- Liu, S., Duffy, A. H., Whitfield, R. I., & Boyle, I. M. 2009. *Integration of decision support systems to improve decision support performance*. Knowledge and Information Systems, 22(3), 261-286.
- Marakas, G.M. 1999. *Decision support systems in the twenty-first century*. Prentice Hall
- McGlinchey, D. 2008. *Bulk solids handling*. Blackwell Publishing
- Micali, V. and Heunis, S. 2011. *Coal Stock Pile Simulation*. Paper presented at Industrial and Commercial Use of Energy (ICUE), 2011
- Mikkelsen, S. 1983. *Various methods of prehomogenising raw materials*. Bulk Solids Handling
- Morrison, J. G., & Moore, R. A. (1999). *Design Evaluation and Technology Transition: Moving Ideas from the drawing board to the Fleet*. In Proceedings of the 5th International Symposium on Command and Control Research and Technology.
- Pavloudakis, F.F. & Agioutantis, Z. 2003. *Simulation of bulk solids blending in longitudinal stockpiles*. International Journal of Surface Mining, Reclamation and Environment, 17(2): 98-112.
- Petersen, I.F. 2004. *Blending in circular and longitudinal mixing piles*. Chemometrics and Intelligent Laboratory Systems, 74(1): 135-141.
- Power, D.J. 1999. *Decision Support Systems Glossary*. [Online]. Available: <http://DSSResources.com/glossary>
- Power, D. J. 2001. *Supporting decision-makers: An expanded framework*. Informing Science, 4(4).
- Power, D.J. 2002. *Decision support systems: Concepts and resources for managers*. Quorum Books
- Rhodes, M. 1998. *Mixing and segregation, Introduction to particle technology*. John Wiley & Sons.
- Robinson, G.K. 2004. *How much would a blending stockpile reduce variation?* Chemometrics and Intelligent Laboratory Systems, 74(1): 121-133.

SACPS. 2011. Materials handling, in England, T., Hand, P.E., Michael, D.C., Falcon, L.M. and Yell, A.D. (eds.). *Coal preparation in South Africa*. Pietermaritzburg: The South African Coal Processing Society.

Sprague Jr, R. H., & Carlson, E. D. 1982. *Building effective decision support systems*. Prentice Hall Professional Technical Reference.

SRI On-line. 2001. *Maths & Decision Systems Group, Silsoe Research Institute*. [Online]. Available: <http://www.sri.bbsrc.ac.uk/scigrps/sg9.htm>

Srivastava, A. 2001. *Model Management*. [Online]. Available: <http://www.gsu.edu/~gs01cpl/ModelManagement.htm>

Stewart, J. 2003. *Calculus, 5th edition*. Belmont, California, USA: Thomson Learning

Van der Mooren, A.L. 1962. *The Homogenisation of Streams of Coal*. Paper presented at Fourth International Coal Preparation Congress.

Walkenbach, J. 2007. *Excel 2007 VBA programming for dummies*. Indianapolis, Indiana, USA: Wiley Publishing

Weisstein, Eric W. 2012a. *Distance*. [Online]. Available: <http://mathworld.wolfram.com/Distance.html> [8 November 2012]

Weisstein, Eric W. 2012b. *Standard Deviation*. [Online]. Available: <http://mathworld.wolfram.com/StandardDeviation.html> [8 November 2012]

Winston, W.L. 2004. *Simulation, Operations research: Applications and algorithms*. Vol. 4. Canada: Brooks / Cole.

Wintz, S. 2011. *Bed Storage and Blending Applied to Gypsum*. Global Gypsum

Wolpers, F.M. 1995. *Homogenization of Bulk Material in Longitudinal and Circular Stockpile Arrangements*. Paper presented at Beltcon - South African Institute of Materials Handling.

## **Appendix A: Simulation results**

Table 13: Simulation results for coneshell model

Coneshell stacking							
	Stacker increment (degrees)	Stacked height (m)	Mean	Min	Max	Variance	VRR
INPUT	-	-	22.65	18.42	27.95	0.831	1
Simulation 1 (base case)	10	17	22.66	21.38	23.65	0.242	0.291
Simulation 2	6	17	22.66	21.4	23.76	0.242	0.275
Simulation 3	8	17	22.66	21.28	23.71	0.246	0.281
Simulation 4	12	17	22.64	21.38	23.67	0.255	0.302
Simulation 5	14	17	22.63	21.33	23.72	0.254	0.305
Simulation 6	10	15	22.65	21.29	23.83	0.314	0.377
Simulation 7	10	16	22.65	21.34	23.75	0.291	0.350
Simulation 8	10	18	22.64	21.38	23.52	0.203	0.245
Simulation 9	10	19	22.65	21.48	23.63	0.173	0.208

Table 14: Simulation results for chevcon model

Chevcon stacking								
	Rotation angle (degrees)	Stacker speed (m/min)	Layers	Mean	Min	Max	Variance	VRR
INPUT	-	-	-	22.63	18.42	25.55	0.761	1
Simulation 1 (base case)	100	25	200	22.65	21.18	23.83	0.175	0.221
Simulation 2	50	25	200	22.64	21.25	23.7	0.278	0.365
Simulation 3	75	25	200	22.65	21.17	23.56	0.225	0.296
Simulation 4	125	25	200	22.67	21.5	23.45	0.126	0.166
Simulation 5	150	25	200	22.64	21.46	23.47	0.133	0.174
Simulation 6	100	20	200	22.61	21.49	23.46	0.174	0.229
Simulation 7	100	22.5	200	22.64	21.43	23.47	0.178	0.234
Simulation 8	100	27.5	200	22.69	21.43	23.51	0.151	0.199
Simulation 9	100	30	200	22.73	21.59	23.46	0.122	0.160
Simulation 10	100	25	100	22.76	21.98	23.55	0.113	0.149
Simulation 11	100	25	150	22.73	21.55	23.45	0.123	0.162
Simulation 12	100	25	250	22.70	21.50	23.57	0.151	0.199
Simulation 13	100	25	300	22.6	21.28	23.5	0.212	0.279
ADDITIONAL SIMULATIONS RUN								
Simulation 14	100	25	20	22.73	21.93	23.43	0.119	0.181
Simulation 15	100	25	33	22.72	21.94	23.54	0.116	0.185

